SUPPLEMENTARY INFORMATION

Clinical Concept Normalization with a Hybrid NLP System Combining Multi-level Matching and Machine Learning Ranking

Long Chen[1,*], Wenbo Fu[1], Yu Gu[1], Zhiyong Sun[1], Haodan Li[1], Enyu Li[1], Li Jiang[1], Yuan Gao[1], Yang Huang[1,*]

[1]Med Data Quest, Inc., San Diego, California, USA

[*]Correspondence to:

Dr. Yang Huang

Post address: Med Data Quest, Inc., 10590 West Ocean Air Drive, Suite 220, San Diego, CA 92130, USA

E-mail address: yanghuang@meddataquest.com

Telephone number: 858-247-5220

Dr. Long Chen

Post address: Med Data Quest, Inc., 10590 West Ocean Air Drive, Suite 220, San Diego, CA 92130, USA

E-mail address: longchen@meddataquest.com

Telephone number: 858-247-5220

Keywords: clinical natural language processing, concept normalization, attention, CUI, UMLS

S1. Definitions of the task-specific matching levels

As mentioned in the main content, each matching level consists of 5 components: Term modification, Dictionary, Query, Matching & ranking, and Disambiguation. And for each component, there are several options or methods to choose from. For the n2c2 task, we started with a simple matching level defined as the match between the given mention and the CUI synonyms. And then error analysis was conducted to tailor the existing matching levels or to implement another matching level. This process (error analysis + matching level tailoring/implementation) was performed recursively until the remaining errors were mostly due to lacking semantic information. Then we developed machine learning ranking systems to deal with the remaining cases. Table S1 shows the definition of each matching level we used for the n2c2 task.

Table S1 Definition of each matching level

Level	Term modification	Dictionary	Query	Matching and ranking	Disambiguation
1	Lower case	Train + UMLS subset	Contain all words	Exact match	majority class from training data
2	Lower case + Stop words removal	Train + UMLS subset	Contain all words	Exact match	majority class from training data
3	Lower case + Lemmatization	Train + UMLS subset	Contain all words	Exact match	majority class from training data
4	Lower case + Medication normalization	Train + UMLS subset	Contain all words	Exact match	majority class from training data
5	Lower case + Abbreviation replacement	Train + UMLS subset	Contain all words	Exact match	majority class from training data
6	Lower case	UMLS full-set	Contain all words	Exact match	majority class from training data
7	Lower case + Stop words removal	UMLS full-set	Contain all words	Exact match	majority class from training data
8	Lower case + Lemmatization	UMLS full-set	Contain all words	Exact match	majority class from training data
9	Lower case + Medication normalization	UMLS full-set	Contain all words	Exact match	majority class from training data
10	Lower case + Abbreviation replacement	UMLS full-set	Contain all words	Exact match	majority class from training data
11	Lower case + Stop words removal	Train + UMLS subset UMLS full-set	Contain at least one word Contain all words	ML ranking	similarity scores of other synonyms instead of the best matching one

Here in table S1, "Train" in Dictionary referrers to the annotated Term-CUI mappings from the training data; "UMLS subset" in Dictionary referrers to the set of synonym-CUI mappings from UMLS that the CUIs are included in the training dataset; "UMLS full-set" in Dictionary referrers to the set of synonym-CUI mappings from UMLS that the CUIs are not included in the training dataset. "Contain all words" in Query stands for the queries requiring the CUI synonyms to contain all words from the mention, while "Contain at least one word" queries only require the CUI synonyms to contain at least one word from the mention.

As established in table S1, level 1-10 correspond to exact match-based matching and level 11 corresponds to machine learning ranking-based matching. More specifically, level 1-5 were designed to find the corresponding CUI directly from the Dictionary of either the annotated Term-CUI mapping from the training data or the UMLS subset (with CUI included in training dataset), regarding different options of Term modification. Level 6-10 were designed similar to level 1-5 but targeted at handling the remaining ones that cannot be found by level 1-5, by searching in a much bigger Dictionary: UMLS full-set. All the remaining ones that cannot be found by exact math (level 1-10) were sent to ML ranking systems (level 11).

In level 11, all the given mentions were processed with lower case and stop words removal. And then, we defined two criteria to fetch the CUI candidates: (1) CUI from the Dictionary containing the annotated Term-CUI mapping from the training data and the UMLS subset, and its synonym must contain at least one word from the given mention; (2) CUI from the UMLS full-set (excluding those in UMLS subset), and its synonym must contain all the words from the given mention. Moreover, cosine similarities between mentions and CUI synonyms represented by average-pooling of the word embedding were used as the default score to rank and to further select CUI candidates. Then the top 15 CUIs from the first criteria (Train + UMLS subset) and top 15 CUIs from the second criteria (UMLS full-set) were selected as the candidates according to their ranking scores (maximum score among all synonyms of each CUI). Then these 30 candidates were sent to ML ranking systems for model training (in training phase) or CUI prediction (in testing phase).

S2. Attention layer

Attention layer provides a trainable weight vector that guides the system to focus on more task-specific semantic information. After the attention layer, word-level features from each timestep are converted to the phrase-level feature vector. For example, by given a phrase consisting of T words $P = \{x_1, x_2, \dots, x_T\}$. After the word embedding, every word x_i transforms to the corresponding word embedding vector e_i, which is a realvalued numerical vector with dimension of d^{wemb} (200 in this study). So after word embedding, the phrase initially as a sequence of words transforms to a sequence of word embedding vectors:

$$E = \{e_1, e, \ldots, e_T\} \tag{s1}$$

The attention layer is a trainable weight vector (w) with the same dimension (d^{wemb}) of the word embedding in this study. Then, the pooling weights of each word can be calculated as:

$$\alpha = softmax(w^T E) \tag{s2}$$

Here, α is the vector of the pooling weights with dimension of T. And w is the trainable attention vector, and w^T is the transpose. Then, the final representation of the phrase can be calculated as the weighted-sum of the word embedding vectors:

$$r = E\alpha^T \tag{s3}$$

Here α^T is the transpose of the vector of pooling weights. And r is the final representation of the phrase with dimension of d^{wemb}.

As shown in equation s2, The pooling weight of each word highly depends on the dot product between the attention vector and the word embedding ($w^T e_i$). As each dimension in word embedding represents a certain semantic feature/axis, the attention layer actually serves as a semantic feature selector which helps the system to focus more on the words carrying more task-specific semantic information. Further discussion upon the effect of the attention layer on the semantic ranking task is included in the Results and Discussion section of the main content, where we provide two examples in figure 4, comparing the semantic ranking results by using average-pooling or attention-based pooling algorithms.

S3. General clinical NLP system

The general clinical NLP system (GCNLP) which employs UMLS and UIMA framework contains 5 main modules: Text processing, Grammar analysis, Entity and relation, Knowledge reasoning and Concept linking.

The first 4 modules are the main modules of the system, while the last one is a task-specific module that was designed to link the GCNLP extracted entities to the n2c2 annotated mentions. In the Text Processing module, all the sub-modules like tokenization, sentence division, section detection use rule/ML hybrid methods and are pre-trained with a much larger dataset. More specifically, the sentence boundary detection is based on maximum entropy classifier[1,2] and rules considering abbreviations, numerical values, date and time, etc. The word tokenization uses a modified Stanford Tokenizer[3] with rules regarding abbreviations. In Grammar analysis, spaCy[4], NLTK[5] are used for syntactic analysis, POS tagging, dependency parsing, etc. The dataset used for training and rules development contains open-access medical data such as MIMIC III[6], and data from previous NLP challenges[7].

In Entity and relation module, the entities such as diseases, disorders, symptoms, medications, procedures are extracted by using modified Lucene[8] lookup algorithm for corresponding Concept Unique Identifiers (CUI) in UMLS. For other entities such as time mentions, lab results, medication dosages, a hybrid NER module combining deep learning (e.g. bidirectional LSTM-CRF models[9,10]), regular expression and lexical/syntactic rules is used. Knowledge graphs (e.g. CUI-CUI relations in UMLS) and deep learning models (e.g. bidirectional LSTM[11]) are employed for relation assignment, including the treatment relations between drugs and diseases, time relations between time mentions and medical activities, etc. A disambiguation submodule based on PageRank algorithm[12] and vector space model[13] is also employed considering context information, semantic type as well as the co-occurrence among concepts. The Knowledge reasoning module then validates/corrects these entities and relations (e.g. merge two concepts to one) base on rules generated from medical knowledge inputs and error analysis to facilitate accurate data analysis. All the machine-learning based models are pre-trained with a much larger medical dataset as mentioned above.

To hook up with the n2c2 task, we linked the n2c2 annotated mention to the NLP extracted entities by calculating their span overlap. For the case one given mention overlapping with multiple entities, the entity with maximum overlap was selected. In contrast to the hybrid systems, we intentionally minimized the finetuning with the n2c2 data for this system. Moreover, it only used UMLS full-set as the dictionary for concept lookup without considering the annotation preference learned from training data.

Reference

1 Reynar JC, Ratnaparkhi A. A maximum entropy approach to identifying sentence boundaries. In: *Proceedings of the fifth conference on Applied natural language processing* -. Morristown, NJ, USA: : Association for Computational Linguistics 1997. 16–9. doi:10.3115/974557.974561

2 Agarwal N, Ford KH, Shneider M. Sentence Boundary Detection Using a MaxEnt Classifier. In: *Proceedings of MISC* . 2005. 1–6.

3 Stanford Tokenizer. https://nlp.stanford.edu/software/tokenizer.shtml (accessed 9 Apr 2019).

4 spaCy. https://spacy.io/ (accessed 16 Jan 2019).

5 Natural Language Toolkit — NLTK. https://www.nltk.org/ (accessed 30 Jan 2019).

6 Johnson AEW, Pollard TJ, Shen L, *et al.* MIMIC-III, a freely accessible critical care database. *Scientific Data* 2016;**3**:160035. doi:10.1038/sdata.2016.35

7 i2b2 NLP Research Data Sets. https://www.i2b2.org/NLP/DataSets/Main.php (accessed 9 Apr 2019).

8 Apache Lucene. http://lucene.apache.org/ (accessed 16 Jan 2019).

9 Lample G, Ballesteros M, Subramanian S, *et al.* Neural Architectures for Named Entity Recognition. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2016. 260–70. doi:10.18653/v1/N16-1030

10 Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging. Published Online First: 9 August 2015.http://arxiv.org/abs/1508.01991 (accessed 9 Apr 2019).

11 Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures.

In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2016. 1105–16. doi:10.18653/v1/P16-1105

12 Agirre E, Soroa A, Stevenson M. Graph-based Word Sense Disambiguation of biomedical documents. *Bioinformatics* 2010;**26**:2889–96. doi:10.1093/bioinformatics/btq555

13 Melamud O, Levy O, Dagan I. A Simple Word Embedding Model for Lexical Substitution. In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2015. 1–7. doi:10.3115/v1/W15-1501