Understanding Medical free text: A Terminology driven approach

Santosh Sai Krishna
gsk.krishna@gmail.com

Manoj Hans
manojhans1989@gmail.com

Abstract

With many hospitals digitalizing clinical records it has opened opportunities for researchers in NLP, Machine Learning to apply techniques for extracting meaning and make actionable insights. There has been previous attempts in mapping free text to medical nomenclature like UMLS, SNOMED. However, in this paper, we analyzed diagnosis in clinical reports by mapping into ICD10 codes. We propose a lightweight approach with real-time predictions by introducing concepts like WordInfo, root word identification. We were able to achieve 68.3% accuracy over clinical records collected from qualified clinicians. Our study would further helps the healthcare institutes in organizing their clinical reports based on ICD10 mappings and derive numerous insights to achieve operational efficiency and better medical care.

1 Introduction

A vast amount of non-standardised clinical reports are available which are rich in information about patient care and disease progression. These clinical reports rarely follow any standards and have minimal grammatical correctness. These reports are usually documented by qualified practitioners about patient's medical history. However, increasing demand for accessing clinical data in industry needs a process for extracting structure and meaning out of the available clinical reports.

There are two major problems for extracting insights from clinical reports. One is unavailability of structured medical data and other is available data is highly varied in terms of terminology for any given phenomenon in the medical field. The main reason for this discrepancy is different information systems in clinics and hospitals. All of these systems have their own separate rules and terminologies to record medical data. This lack of consistency between data from different information systems has reduced interoperability across health care organisations. In order to improve interoperability, the data must be represented using standard terminologies.

The present work proposes a goal to develop a system of mapping free text such as patients clinical report and diagnosis with an ICD-10 code for a disease.

2 Related Work

Some known systems for mapping free text to UMLS are SAPHIRE (Hersh et al., 1995), MetaMap (Aronson, 2001), IndexFinder (Zou et al., 2003), and NIP (Huang et al., 2005). The SAPHIRE system uses a lexical approach and maps text to UMLS terms. Later, IndexFinder added Semantic and Syntactic filtering to improve performance of lexical mapping. NIP uses sentence boundary detection, noun phrase identification and parsing, all of these are computationally expensive processes.

MetaMap is another approach to map free text to a terminology like UMLS. This approach uses a three step process, where a free text is first broken down to simple noun phrases using the Specialist minimal commitment parser. After this, variant of phrases and mapping candidates are generated using UMLS source vocabulary. Then for all of these candidates, a score is generated to evaluate the best fit

Proceedings of the 5th International Workshop on Computational Terminology,
pages 121–125, Osaka, Japan, December 12 2016.

medical concept of each term. MetaMap is also computationally expensive therefore unsuitable for real time processing.

The work done by (Hazlehurst et al., 2005) is on mapping free text to UMLS by generating all the synonyms of each word of the input. All these words and synonyms are used to find the best possible combination among them, which matches a concept in UMLS. This process matches 1 concept every 20 seconds or longer thereby unsuitable for real-time concept mapping.

SNOMED CT also offers a huge potential for standardising clinical reports into medical concepts. One of the known research by (Patrick and Wang, 2007) uses augmented lexicon, term composition and negation detection to come up with phrases that have a potential match of concepts from SNOMED CT. The idea is to come up with medical concepts which can properly describe a given clinical note. The major limitation for this approach is that it is not considering various possible order of words that can be used while writing a report. It is expecting user to write reports according to the rules and standards used in SNOMED CT.

We are offering an approach which considers multiple combination of noun phrases, ordered by its entropy and which can be used in free writing. It is computationally inexpensive and can be used in real time systems to map free text to a disease code in ICD-10

3 Architecture

Given a clinical report on a patient history, we need to map the diagnosis into ICD10 code(s). As the clinical reports are filled by Doctors, who have numerous reports to fill in a day, there are chances of human errors in spelling variants and sometimes the order of words. Before we even map the individual words to the most descriptive ICD10 code, we need to clean and normalise the words. Furthermore, the resulting ICD10 codes needs to be ranked based on their relevancy to the diagnosis and also the irrelevancy with the remaining words in a ICD10 code. All the above steps put together fall into a pipelined approach as detailed below:

3.1 Preprocessing

With any error in spelling it would be easy to miss the optimal ICD10 code. If someone types “acute gastrtis“ instead of “acute gastritis“, then we would be left with ICD10 codes that match only the word “acute“. We lost the primary context of “gastritis“ and this results in a misleading classification to “acute“. Hence, it is crucial to resolve the spelling mistakes. To resolve the misspelling, all unique words mentioned in all ICD10 codes are collected and a Trie data structure is built on the characters of each word. The resulting spell correction algorithm is able to suggest correct words in less than 1ms with a maximum edit distance of 2. After spell correction, the text is cleaned by removing non alphanumeric characters, any ICD codes and later followed by lemmatization.

3.2 Finding Primary and Secondary words

To match a diagnosis with an ICD10 code it is often difficult and not necessary to match the entire diagnosis text, it would be enough to match the context, e.g., “dengue fever“ can be matched with a concept having “dengue“ rather than looking for both the words to be matched. This means that we need to identify the root word for every medical word. In our example, “dengue“ is_a “fever“ . If such a mapping can be derived, then every diagnosis free text can be split into two sets of words, i.e., primary (must match) and secondary (should match). The primary list follows a must match criteria whereas the secondary list doesn’t follow a strict criteria however any code that contain words from the secondary list will be ranked higher.

- **Deriving root words** : Understanding the terminology of ICD10 gives us a great context of the organisation of diseases. For example, the concept “K12“ which talks about “stomatitis (inflammation in mouth)“ has its ancestors concept as “K00-K14“ that elaborate “Diseases of oral cavity, salivary glands and jaws“. This ancestor mapping provides an insight of hypernym relation (root words) like “stomatitis“ is_a “disease“, “stomatitis“ is_a “oral cavity“, etc. Any hypernym relation which appears above 80% of all ancestor mappings is considered as a root word.

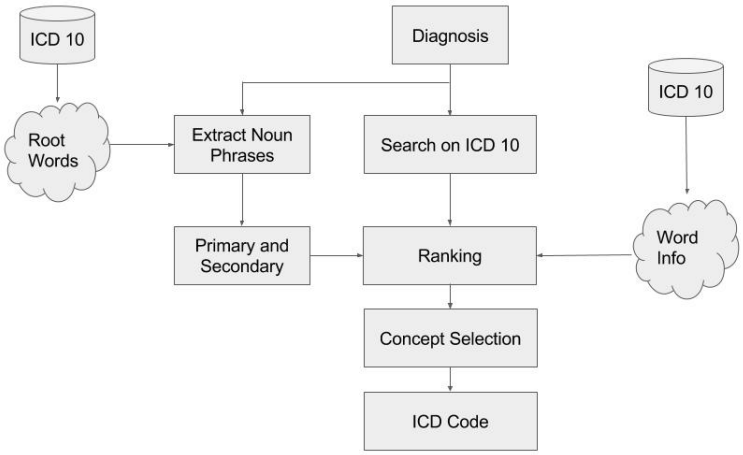

Figure 1: Architecture Components

- **Finding the Primary/Secondary split** : With the diagnosis text preprocessed, all noun phrases are extracted using the nltk package in python. Within each noun phrase, any possible root words will be extracted using the mappings extracted from the above step. All the root words are considered as Secondary and all the specific (non-root) words are considered as Primary. The remaining words of the diagnosis are added to the Secondary list. For example, "dengue" (non-root) would map to "fever" (root), which makes "dengue" as primary and "fever" as secondary.

3.3 Word Informativeness

You shall know a word by the company it keeps (Firth, J. R.). If a word like "other" is often seen and with different words, then it carries very less information, compared to a word like "cardiac" which is often used in the very specific context of heart-related diseases. The WordInfo is a metric derived from the randomness associated with a word. It defines how random a word is based on the number of unique words seen in its surrounding context. This helps in ranking ICD10 codes when multiple codes are matched with the terms in diagnosis. It derives from the idea that an ICD10 mapping should be as specific about the diagnosis as possible and as generic as possible in terms of the remaining words of that code.

The WordInfo is calculated in two steps as detailed below:

- **Calculating PMI scores** : The PMI scores are calculated using the below formula

$$PM1(x, y) = \log(\frac{P(x, y)}{P(x).P(y)})$$

  The above equation helps us in deriving the pointwise mutual information between any two specific words. A higher PMI score indicates a closer association between two words. However, we want to capture the randomness of an individual word.

- **The Skewness of PMI scores** : For common words like "other", there would be high number of co-occurring words with varying PMI scores. We had assigned WordInfo to be the third moment (skewness) of PMI scores because we have observed that common words share a skewness below zero compared to words like "cardiac". This is attributed to the fact that a number of infrequent words are being associated with common words and hence resulting in a left-skewed distribution of PMI values.

3.4 Search and Rank

An inverted index is built over all ICD10 codes. A boolean OR query is performed on the tokens extracted from a diagnosis to retrieve all ICD10 codes that mention one or more of the diagnosis tokens. The retrieved concepts need to be ranked based on their relevancy to the diagnosis. Ideally, we want the retrieved codes topic to be matching with the diagnosis and doesnt contain any other topic. Based on these two factors, the ranking algorithm is designed as follows

- **Likelihood of unique Primary words** : After segregating a diagnosis text into primary and secondary, the extent of context overlap between the diagnosis and the code can be identified with the likelihood of finding unique Primary words in the code terminology.

$$P(\text{Unique Primary}) = \frac{\text{No. of unique primary matched—}}{\text{No. of total unique primary words in query}}$$

- **Total likelihood of Primary words** : This measures the probability of finding a Primary word in the definitions of a Code.

$$P(\text{Total Primary}) = \frac{\text{No. of primary words matched}}{\text{No. of total words in a code}}$$

- **Likelihood of unique Secondary words** : After having observed the Primary words, this measure evaluates the relevancy of a Code based on the overlap of Secondary terms.

$$P(\text{Unique Secondary}) = \frac{\text{No. of unique secondary words matched}}{\text{No. of total unique secondary words in query}}$$

- **Total likelihood of Secondary words** : Among the remaining terms, the probability of visiting a Secondary word is calculated with this measure.

$$P(\text{Total Secondary}) = \frac{\text{No. of secondary words matched}}{\text{No. of total words in a code}}$$

- **Randomness associated with the remaining non-query words** : To evaluate if a Code is elaborating other concepts along with the diagnosis we need to understand the information provided by the non-query words. Using the WordInfo metric calculated for each word as described in section above, we will be able to call out the codes that do not match the context in a diagnosis.

3.5 Concept Selection

The retrieved Codes are ranked based on a linear weighted combination of above metrics with weights being assigned on a descending priority of above order. However, the task is to assign Code(s) to a diagnosis. This assignment problem is achieved by a greedy approach. The top ranked Code will be first assigned to a diagnosis. All the Primary words appeared in the assigned Code are removed from the diagnosis and the assignment step is repeated until all Primary words are found in the assigned Code(s).

4 Results

It is hard to compile data sources providing the gold standard of mapping for a given free text. However, we evaluated our algorithm using a dataset gathered from clinical hospitals. The dataset consists of clinical reports written by qualified professional doctors after examining a patient mentioning the case history, diagnosis with its corresponding ICD10 code and a few other details. We extracted diagnosis with its ICD10 code for our evaluation study. Out of the 5823 total case reports available, there were 4902 records that had a non-empty diagnosis and ICD10 code.

As the entire architecture is built on a knowledge base approach which does not rely on supervision of ICD10 codes, we used the entire 4902 samples for evaluating the algorithm. The evaluation metrics

	Elasticsearch	Preprocess + Search + Primary/Secondary	+ WordInfo
Accuracy	25.5	64.2	68.3

used here is Accuracy. We compared the results against a basic approach which used search and ranking capabilities of Elasticsearch.

As per the results mentioned in Table 1, it is evident that our approach is performing better than a basic search algorithm. It can also be seen that the impact of WordInfo scores introduced in this paper is significantly improving the results. With all modules included, our algorithm was able to assign an ICD10 code for a diagnosis with an average time of 25ms.

5 Conclusion

It has been an acknowledged fact that understanding clinical records is crucial in improving the medical care. This paper is an attempt at understanding the diagnosis provided by qualified doctors by mapping them to a standard nomenclature like ICD10. We were able to achieve an accuracy of 68.3% over 4902 records. These results can be attributed to algorithms introduced like the identification of root words, deriving WordInfo values along with a probabilistic ranking approach. However, this work can be further extended by adding synonyms to diagnosis terms, or improving their representation using deep learning models like word2vec, GloVe and also by expanding any abbreviations.

References

Patrick J, Wang Y, Budd P. 2007 *An automated system for conversion of clinical notes into SNOMED clinical terminology.* Proceedings of the 5th Australasian Symposium on ACSW Frontiers, Ballarat, Australia

Hersh, W. R. and D. Hickam 1995 *Information retrieval in medicine: The SAPHIRE experience.* Journal of the American Society for Information Science 46(10): 743-747

Aronson, A. R. 2001 *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.* Proc AMIA Symp 17: 21.

Zou, Q., W. W. Chu, et al. 2003 *IndexFinder: A Method of Extracting Key Concepts from Clinical Texts for Indexing.* Proc AMIA Symp 763: 7.

Huang, Y., H. J. Lowe, et al. 2005 *Improved Identification of Noun Phrases in Clinical Radiology Reports Using a High-Performance Statistical Natural Language Parser Augmented with the UMLS Specialist Lexicon* American Medical Informatics Association.

Hazlehurst, B., H. R. Frost, et al. 2005 *MediClass: A System for Detecting and Classifying Encounter-based Clinical Events in Any Electronic Medical Record* American Medical Informatics Association.