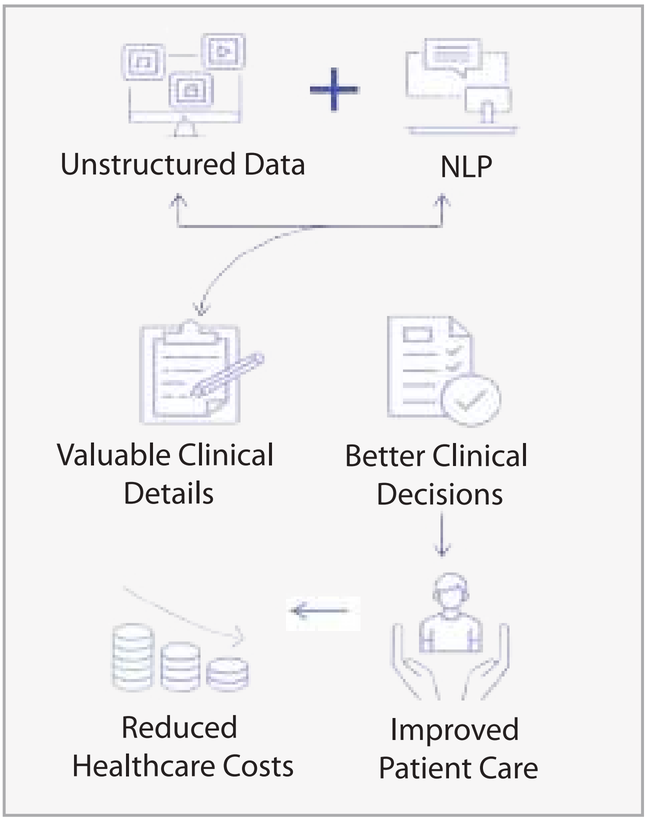
TCG DIGITAL

WHITE PAPER

Tapping into the potential of Unstructured EHR Data

A TCG Digital Perspective

Contents



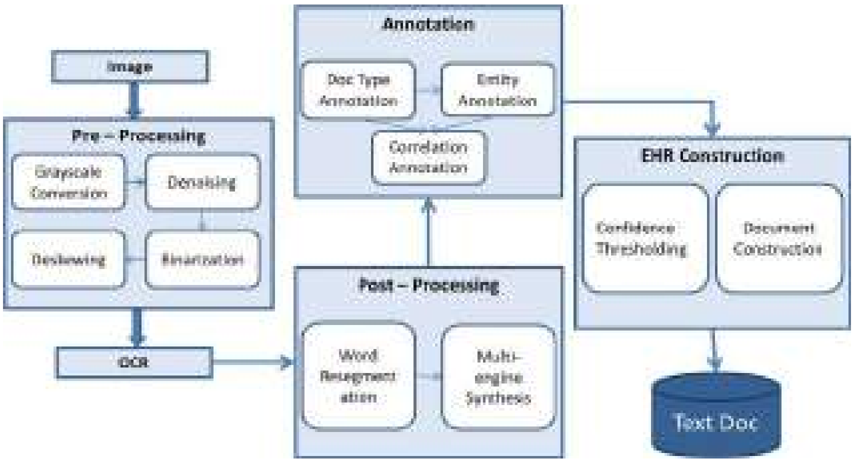
The wide adoption of Electronic Health Record (EHR) frameworks in healthcare produces enormous certifiable information that opens up new possibilities to direct clinical research. Data in EHR can be divided into three kinds: structured data, semi-structured data, and unstructured data.

Structured data, which is generally stored in fixed-mode databases, contains basic information (such as birth data and nationality), drugs taken, allergies, and vital signs (such as height, weight, blood pressure, and blood type,). Semi-structured data usually has the flow chart format, similar to RDF (resource description files), including name, value, and time-stamp.

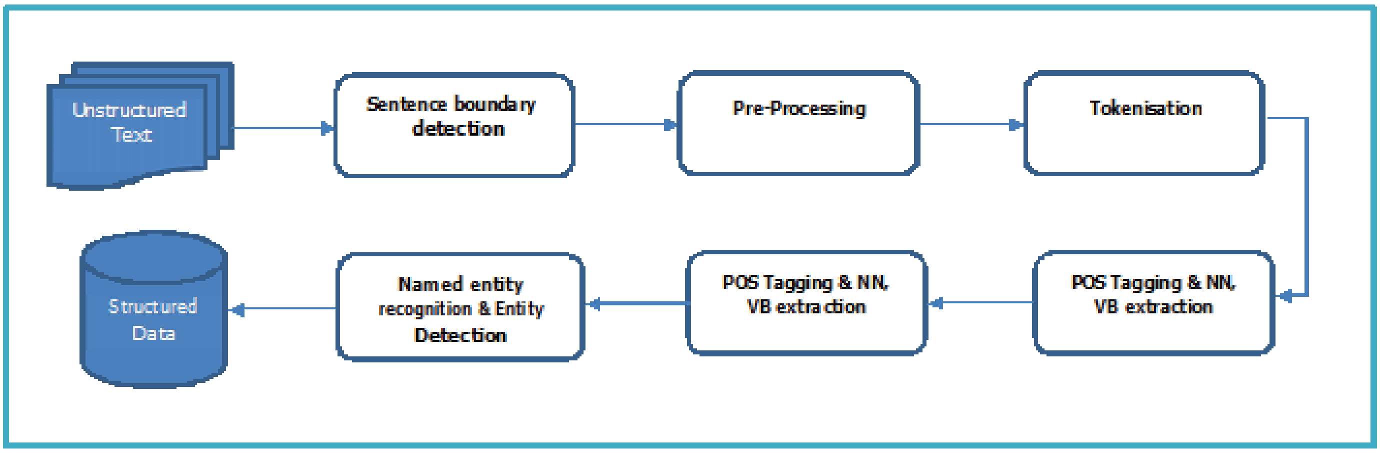
Unstructured text is one kind of narrative data, including clinical notes, surgical records, discharge records, radiology reports, and pathology reports. Unstructured texts store a lot of valuable medical information but lack common structural frameworks, and there are many errors, such as improper grammatical use, spelling errors, local dialects, and semantic ambiguities, which increase the complexity of data processing and analysis. Sometimes even the data available is not digitized properly. For example: the clinical notes or prescriptions are hand written and stored as an Image.

In this paper, we are proposing an approach that can be used to tap the unseen potential of this unstructured data available. A text analytics solution is proposed to help illuminate how using unstructured text data from EHR and medical notes trained on a deep learning NLP network can lead to quicker and powerful insights. The capability promises to extract useful data from EHRs and medical notes and enhance their ability to improve costs, efficiency, and productivity.

Before going to the text analytics and NLP part, first let us understand how we can convert digital or handwritten documents to text using the OpenCV library and existing OCR engine. For the purpose of this paper, we have tried to summarize the overall approach in the diagram below.



Let us now move to the final text analytics approach. We will discuss each step used in the approach


Unstructured Text
Sentence boundary detection
Pre-Processing
Tokenisation
POS Tagging & NN, VB extraction
POS Tagging & NN, VB extraction
Named entity recognition & Entity Detection
Structured Data

1. Sentence Boundary Detection

In this step, the sentences from clinically rich texts are identified. The sentences can be detected by using sentence terminators like period (.), question mark (?) etc.

The period appearing in 5.8 is not considered as sentence boundary rather the actual boundary at the end of sentence is detected as sentence boundary and sentences are separated based on that.

Input	Output
[“FBS & hgA1c both slightly improved, but still pre diabetes (HgA1c = 5.8%). But did instruct on diet/exercise.”]	*[“FBS & hgA1c both slightly improved, but still pre diabetes (HgA1c = 5.8%).”, “But did instruct on diet/exercise.”]*

2. Pre-processing

Pre-processing, as the name suggests, is performed prior to the actual processing and is essential in standardizing the sentences so that it becomes easy in further steps. Following tasks are achieved in this pre -processing phase:

Abbreviation Handling	Punctuation Handling	Lower Case Conversion	ASCII Character Removal
Abbreviated texts are replaced by its full form. A list based replacement approach can be used for this. For example, pt is expanded as patient; dx is expanded as diagnosis etc.	*The texts having punctuation and denoting negation, like “don’t”, “hasn’t” etc. should be converted into actual negative form like “do not”, “has not”. This makes the negation handling part easy to detect the negative scenarios.*	*In order to bring standardization and reduce the case conversion effort during named entity detection phase, all texts can be converted in lower case characters.*	*Since the notes are maintained in different systems, there is chance of having different ASCII characters which makes the program to fail. So ASCII characters should be removed before further processing*

3. Tokenization

Input	Output
[“FBS & hgA1c both slightly improved, but still pre diabetes (HgA1c = 5.8%).”, “But did instruct on diet/exercise.”]	['FBS', '&' , 'hgA1c', 'both', 'slightly', 'improved', ',', 'but', 'still', 'pre diabetes', '(', 'HgA1c', '=', '5.8', '%', ')', '.', 'But', 'did', 'instruct', 'on', 'diet/exercise', '.']

In this step, each sentence is further broken down into individual tokens. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. The python library NLTK can be used for this step. Following is an example of tokenizer – This phase constructs very effective input for next phase which can deal each token wise rather having to deal with large sentences. It can deal with small chunks only.

4. Parts-Of-Speech Tagging

The next step in the processing is to assign a parts of speech tag to each token. This step is required so that we can limit our search to only those tokens which are tagged as Noun or Verb during further phase of Named Entity Recognition. Developed at the University of Pennsylvania, the Penn Treebank tagger is used for POS tagging. POS-tagging algorithms fall into two distinctive groups:

Rule-Based POS Taggers - Typical rule-based approaches use contextual information to assign tags to unknown or ambiguous words.

Input	Output
['FBS', '&' , 'hgA1c', 'both', 'slightly', 'improved', ',', 'but', 'still', 'pre diabetes', '(', 'HgA1c', '=', '5.8', '%', ')', '.', 'But', 'did', 'instruct', 'on', 'diet/exercise', '.']	[('FBS', 'NNS') ('&', 'CC') ('hgA1c', 'NNP') ('both', 'DT') ('slightly', 'RB') ('improved', 'VBN') (',', ',') ('but', 'CC') ('still', 'RB') ('pre diabetes', 'VBZ') ('(', ':') ('HgA1c', 'NNP') ('=', ':') ('5.8', 'CD') ('%', 'NN') (')', ':') ('.', '.') ('But', 'CC') ('did', 'VBD') ('instruct', 'NN') ('on', 'IN') ('diet/exercise', 'JJ') ('.', '.')]

Disambiguation is done by analysing the linguistic features of the word, its preceding word, its following word, and other aspects.

Stochastic POS Taggers - The simplest stochastic taggers disambiguate words based solely on the probability that a word occurs with a particular tag. In other words, the tag encountered most frequently in the training set with the word is the one assigned to an ambiguous instance of that word.

5. NN-VB Extraction

Once the token is tagged, the main area of concern for further processing would be Noun and Verb phrases. So before the actual recognition of entities like diagnosis, procedure etc., we first extract the Noun and Verb phrases in this phase. This makes recognizer module work on small set of significant data only rather than searching through entire dataset. Only the tokens with following tags are extracted from POS tagger output.

Nouns generally refer to people, places, things, or concepts. The simplified noun tags are N for common nouns like book, and NP for proper nouns.

- **Noun phrases -**
  - NN noun, singular
  - NNS noun plural
  - NNP proper noun, singular
  - NNPS proper noun, plural

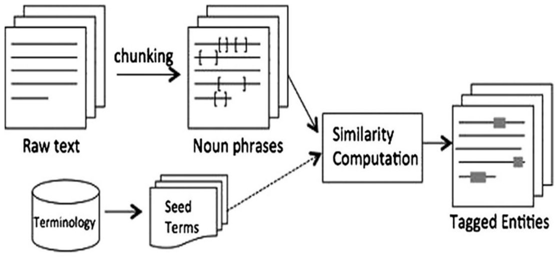
Verbs are action words, looking for verbs the following phrases are used

- **Verb phrases - VB, VBD, VBG, VBN, VBP, VBZ)**
  - VB verb, base form
  - VBD verb, past tense
  - VBG verb, gerund/present participle
  - VBN verb, past participle
  - VBP verb, sing. present, non-3rd
  - VBZ verb, 3rd person sing. present

6. Named Entity Recognition (NER)

Named Entity Recognition (NER) in the healthcare domain involves identifying and categorizing disease, drugs, and symptoms for bio surveillance, extracting their related properties and activities, and identifying adverse drug events appearing in texts. These tasks are important challenges in healthcare. NER process, defines the boundaries between common words in terminology in a particular text, and assigns the terminology to specific categories based on domain knowledge.
In order to effectively implement this module, we have to build a medical corpus with training data set. Later on, the corpus is used to find out various entities through the recognition process. One has to have proper medical ontologies/dictionaries developed before this step.

- **Medical Corpus:**
  The sole purpose of the corpus building is to apply it in Named Entity Recognition phase to classify the correct entity. The idea is to generate a rich tagged set of corpus from the available set of data. This corpus will be then used to tag the unstructured text automatically by the system. For the purpose of this research, we need to identify the entities like Diagnosis, Procedure and Drug, etc. Following are the steps for the algorithm used:
  - Medical records consisting of tests conducted, patient's health status, diseases and response to the treatments are taken as input
  - Concepts like medical tests, diagnosis and treatments mentioned in the clinical records are classified into categories.
  - The records are divided into training data and testing data. 70% of data is used as training data and it is fed to the model.
  - Testing data (30% of data) that consists of patient's information are fed to the model.
  - The real data (clinical records) are fed to the pre developed model.
  - The output includes list of words that indicate test conducted, problem diagnosed or treatment given.



- o From the list of diseases and test conducted, the specializations are classified and displayed.

Other approach is to use Medical dictionaries available online such as RxNorm, SNOMed, LOINC, ICD Mappings, etc.

- **Redundancy Handling:** While going through such huge amount of data collected, we end up having same elements in the corpus repeated several times in the file bringing redundancy in corpus file. Duplicate values should be removed to increase code efficiency.

7. Entity Detection

After the corpus generation, actual entity detection phase is entered using the corpus file. Input to this module is noun-verb extractor output, i.e., Noun and Verb phrases only and the output is recognized entities. Each NN-VB phrase is matched against all corpus files to categorize the element. The extracted entities would be processed further downstream to link entities and leverage dictionary-based techniques for flagging any symptoms which could potentially be adverse drug reactions to the prescribed medicines. The Named Entity Recognition models built using deep learning techniques extract entities from text sentences by not only identifying the keywords but also by leveraging the context of the entity in the sentence. Furthermore, with language model pre-trained embeddings, the NER models leverage the proximity of other words which appear along with the entity in domain-specific literature.

7. Entity Detection

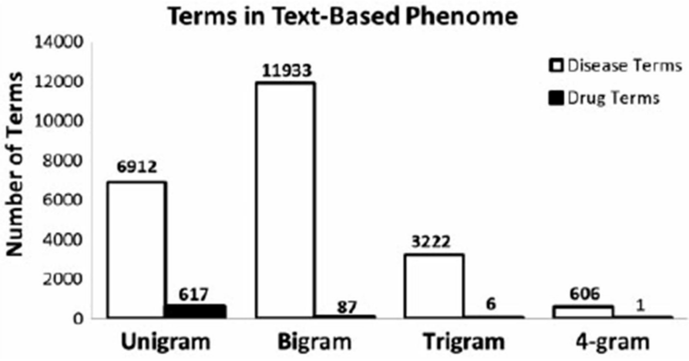
Some other components that can be integrated to the systems:

- **Negation handlings:**
  Negation handling is an automatic way of determining the scope of negation and inverting the polarities of opinionated words that are actually affected by a negation. The negative words like no, none, free etc. are tracked and flag the detected entity as negative so that true negative rate is good.
  Examples of Negation
  - o Not diabetic.
  - o No chest pain.
  - o No weight loss or episodes of stomach pain.
  - o Hypertension absent.

  Context — this refers to a condition that a patient had previously or a relevant condition that the patient's family member had.
  Examples of Context
  - o Patient's mother and father developed Diabetes in their 50s.
  - o Patient — long history of common cold.

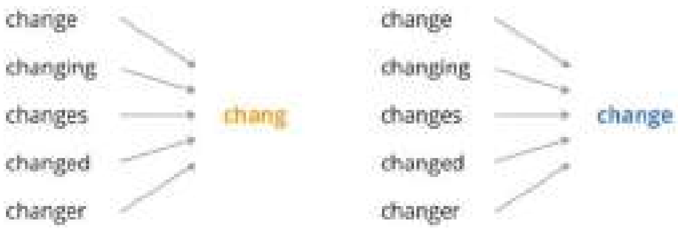
Negation handlings:

N-grams analyses are often used to see which words often show up together. It is often better to investigate combinations of two words or three words, i.e., Bigrams /Trigrams, in which the consecutive two words are analysed in unison to detect more positive entities like heart attack, liver disease etc.



Stemming & Lemmatisation:

In which the stem of word is extracted and compared so that we have less redundancy and better performance.



Once, the data gets into structured format, the data can be used for other applications:

Some applications or use cases:

Clinical Trial Matching – The data can be structured using this approach for the Patients available as well as for the trial eligibility criteria. Having structured data, the patients can be easily matched to the clinical trials.

Clinical Decision Support – Natural language processing and machine learning in healthcare enable healthcare professionals to make better decisions. Certain areas in healthcare need better methods of surveillance, such as medical errors. NLP is also being used to aid clinicians in checking symptoms and diagnosis.

Risk Adjustment and Hierarchical Condition Categories – Most of the times patients pay more than what is needed. Hierarchical Condition Category coding, a risk adjustment model, was initially designed to predict the future care costs for patients. In value-based payment models, HCC coding will become increasingly prevalent. Natural language processing can help assign patients a risk factor and use their score to predict the costs of healthcare.

Identify patients with critical care needs – NLP algorithms can extract vital information from large datasets and provide physicians with the right tools to treat patients with complex issues.

Root Cause Analysis and Predictive Analytics – Applying NLP to vast caches of electronic medical records can help identify subsets of geographic regions, ethnic groups or other population segments that face different types of health disparities.

Ambient Virtual Scribe – It is an idea for the future implementation as a better solution of manual entry of data into EHRs. The system works through microphones in the examining room that record the conversation between the patient and doctor. The NLP kicks in by automatically transcribing the conversation into an EHR for follow-up treatments.

Summary

We have sought to provide a broad outline of the current state-of-the-art, opportunities, challenges, and needs in the use of NLP for handling EHR, with a particular focus on unstructured data. We have provided a brief on the architecture which can enable us to handle handwritten notes and convert them to text document. Furthermore, we have outlined methodological aspects from a clinical as well as an NLP perspective and identify seven broad steps for structuring and extracting data from clinical reports. Based on these, we provide actionable guidance for each identified step. We envision further advances particularly in methods for data processing methods that move beyond current basic techniques and move closer to clinical practice and utility, and in transparent and reproducible method development. We have also provided insights for various use cases where we can leverage the framework mentioned in this paper, which can help in multiple medical avenues

Conclusion

NLP in healthcare is still at a nascent stage. But the industry is eager to make strides in the effort. Semantic big data analytics and semantic processing ventures of NLP foundations are seeing major healthcare investments from some recognised players. In healthcare and life sciences, the global NLP market size will rise from $1.5 billion in 2020 to $3.7 billion by 2025, with a CAGR of 20.5%.[13]

Healthcare organizations are already using NLP to get at the low-hanging fruit, and major tech entities are leveraging NLP in health-related tools; Amazon, for example, recently released a user-friendly clinical NLP [14] tool. Many open-source tools are available at no cost—allowing users to do classification, find phrases, and look for contextual information that provides clues about family history. But to maximize NLP's potential in healthcare; organizations need to look beyond these off-the-shelf solutions to healthcare-specific vendor systems that integrate into existing workflows.

References

1. https://www.binaryfountain.com/blog/use-cases-nlp-in-healthcare/
2. https://marutitech.com/use-cases-of-natural-language-processing-in-healthcare/
3. https://www.foreseemed.com/natural-language-processing-in-healthcare
4. J. Perkins, "Python Text Processing with NLTK 2.0: Creating Custom Corpora", URL-https://www.packtpub.com/books/content/python - text-processing-nltk-20-creating-custom-corpora, November 2010 (visited on August 2015)
5. A. Coffman and N. Wharton, "Clinical Natural Language Processing Auto-Assigning ICD-9 Codes", 2007
6. https://www.researchgate.net/publication/323282101_A_new_approach_to_extract_meaningful_clinical_information_from_medical_notes
7. https://www.imaginea.com/the-rise-of-nlp-in-the-healthcare-industry/#:~:text=In%20healthcare%20and%20life%20sciences,with%20a%20CAGR%20of%2020.5%25.&text=It%20is%20up%20to%20the,b ased%20care%2C%20billing%20or%20workloads.
8. https://www.healthcatalyst.com/insights/how-healthcare-nlp-taps-unstructured-datas-potential
9. https://www.healthcatalyst.com/insights/healthcare-nlp-4-essentials
10. https://www.ohdsi.org/data-standardization/the-common-data-model/
11. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6372467/
12. Andreea Bodnari, Louise Deleger, Thomas Lavergne, "A Supervised Named-Entity Extraction System for Medical Text"
13. https://www.coherentmarketinsights.com/market-insight/natural-language-processing-nlp-in-healthcare-and-life-sciences-market-2798
14. https://aws.amazon.com/comprehend/medical/
15. http://support.ptc.com/help/mathcad/en/index.html#page/PTC_Mathcad_Help/thinning_and_skeletonizing.html
16. https://www.pyimagesearch.com/2017/02/20/text-skew-correction-opencv-python/
17. https://medium.com/technovators/survey-on-image-preprocessing-techniques-to-improve-ocr-accuracy-616ddb931b76
18. https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7
19. https://towardsdatascience.com/image-filters-in-python-26ee938e57d2
20. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, p. 436, 2015.
21. https://towardsdatascience.com/transformer-neural-network-step-by-step-breakdown-of-the-beast-b3e096dc857f
22. https://towardsdatascience.com/transformers-141e32e69591
23. https://arxiv.org/pdf/1904.03323.pdf