Natural Language Processing and Web Tools for Mapping Units from ClinicalTrials.Gov

Jacob Barhak
Austin, TX 78758
734-998-3737
jacob.barhak@gmail.com

Joshua Schertz
San Antonio, TX 78232
210- 621-8000
joshschertz3@gmail.com

Keywords:
NLP, Clinical Trials, Units, Modeling, Web Site

ABSTRACT: *Summary data from some clinical trials is now becoming available in electronic format due to US public law through the NIH NLM database ClinicalTrials.Gov . The database currently holds over a quarter of a million trials with about 30K trials with result records. However, despite the great work of the team that developed this fast growing database, the data held in it is far from standardized and using it requires effort, especially for machines. The difficulty arises from the fact that data entry to this database is manual from multiple external sources, mostly textual, and somewhat permissive. Although entered data goes through a review process, the review is human and therefore sometimes forgiving. For machine comprehension, definition of units is essentials so numbers held in the database will make sense. There are currently over 20K units for 30K clinical trials with results and many of the units are synonyms and some are even errors. Even CDISC units that are in a good level of standardization need normalization and enhancement. This presentation will discuss recent advances in the effort to standardize the medical units. Specifically Natural Language Processing (NLP) techniques are used alongside other Machine Learning methods to cluster similar units together. This allows a human to inspect and standardize the units more efficiently. In this presentation the NLP techniques used will be discussed in details as well as the clustering techniques. Similar techniques merged with web tools will be helpful for future analysis of other textual fields within the fast growing database. To move the standardization effort forward a web tool was created to allow humans to classify the units. Multiple users can see the units in each cluster and the machine suggestions and classify those. The web portal is accessible through: ClinicalUnitMapping.com .*

1. Introduction

The practice of medicine was dominated by human decision for centuries. Tools such as philosophical reasoning [1] and the scientific method exercised in clinical trials were used to support human reasoning. Human reasoning relies on individual knowledge and experience of the physician supported by guidelines, which summarize group knowledge, and availability of literature to allow seeking new knowledge. However, the amount of medical knowledge accumulated and being generated has reached a point where a the knowledge generated is much more than any individual can absorb. According to [2] a new medical article appears every 26 seconds, and this is a lower bound estimate. Therefore, it is impossible for a single human to even read all this generated knowledge. However, computers can easily store and access this knowledge. In fact, modern search engines and libraries are already online resources. Therefore, the idea of computer reasoning is gaining momentum.

This fact remained unnoticed and IBM Watson [3] attempted to teach medical knowledge to a computer. At the time the claim was that the computer was behaving equivalent to a third year medical student [3] and becoming better. The ability of the computer to process text was clearly demonstrated by IBM with the IBM Watson winning the Jeopardy TV game show [4]. And this task seems to be as complex as other endeavors of computers replacing human driving tasks in self-driving cars [5]. However, although computers can make some tasks much faster and accurate than humans, they rely on clean data with unambiguous terminology and on good definitions to produce good results.

Unfortunately medical text is very ambiguous. In fact, the medical terminology problem is widely addressed through multiple solutions and standards starting from International Disease Classification (ICD) [6], Snomed CT [7,8], Clinical Data Interchange Standards Consortium (CDISC) [10], and those standards are being absorbed to UMLS [11] which absorbs many other standards. While those standards deal with codes, text, and relationships among entities, very little effort was invested in comprehending the numerical part.

However, when clinical trials are conducted, their results are provided as numerical statistics. Typically the baseline population statistics are provided and the outcome statistics at the end of the trial are also provided as tables. Numerical data is easier for a computer to process than for humans if presented properly, yet the lack of standardization results in different organizations and researchers reporting data in different ways. More specifically, results are reported with different units, scales, structures.

So far, humans were reading those papers and drawing conclusions from those, so the situation was manageable, partially due to human limitation on speed. However, with new technologies such as disease models that accumulate knowledge, [12,13] there is a need to fix the standardization problem. The problem became very apparent when clinical trials started accumulating data in ClinicalTrials.Gov .

1.1 ClinicalTrials.Gov

Medical data availability in electronic formats is increasing. This growth allows better analysis of more data, which in turn increases our understanding of medical phenomena. ClinicalTrials.Gov is an important database that provides a look into a large number of medical phenomena [14,15]. It holds more than a quarter million trials where roughly 10% of them publish the results in this database, and this database has been growing fast since its creation [16]. It is now a law for some clinical trials to register their results within this database [17] - compliance to the law started April 18, 2017. This aggregated knowledge base is valuable since it merges data that would otherwise be scattered or inaccessible to researchers. The summary results reported in the database bypass data restrictions imposed on individual data in many cases. Therefore, it allows accumulation of knowledge in ways not possible before. Moreover, since each trial reports on multiple people, and since many trials are reported, phenomena can be potentially researched globally on a larger scale.

However, to employ this great potential, there is a need to interpret the data stored in ClinicalTrials.Gov. Since no strict standard is enforced upon data entry, using the database to gain insights from merged data requires interpretation and pre-processing.

Standardization of this database will allow humans and machines to read the data in a way that allows comparison and further analysis. Data standardization will support human tasks such as systematic review and is essential to make the aggregated data machine readable for modeling and simulation [19]. Yet even now, users can use the database in various ways.

Users of this database can search for trials using multiple criteria and download results in multiple format. In fact the entire database can be downloaded in one archive of xml format files. Each file contains a clinical trial described by a subset of fields represented by xml tags. There are roughly 400 different fields in this database, most of them are textual fields, while some have numeric value. Six fields are dedicated to code units. Those unit fields are the focus in this work since these field are essential to comprehension on numeric components within the database; otherwise, it would be impossible to properly scale numbers from different trials. However, the need for standardization requires much more effort.

1.2 The need for standardization

For example, not all trials have their acronyms entered and the “brief description” field should be consulted to figure out the trial name. In many trials, result cohorts and baseline population cohorts are not matching or redundantly defined. In some cases units are misspelled. For example, the number of individuals participating can be found in the data spelled as: "participants" , "Participants" or "Participant". Another example is BMI units that can be found spelled as: "kg/m²", "kg/m^2", or "kilogram per square meter (kg/m^2)". This unfortunate condition arises from the intentions of designers to make the data entry easier for humans to cope with, so textual input was allowed for many fields, which results in the conditions described above. It also results in unfortunate examples where study length is described in different units and numbers. Examples include: "4.9 years", "five years", "From randomisation to individual end of observation, up to 4.6 years", and "Time from randomization to the first event (Maximum 50 months) Percentage of Participants With Occurrence of Secondary Cardiovascular Composite Endpoint (Core: Active Treatment Phase)". Such examples require some text processing to allow such data to be compared and reused in analysis. However, some cases are harder to interpret and may need manual human treatment. The following entry of study length in an example: "Until a primary cardiovascular event (death, myocardial ischemia, congestive heart failure, myocardial infarction or cerebrovascular accident) occurred or 28 March 2009, whichever occurred first" – this entry either requires human intervention of much deeper understanding of the data by the machine since the field contains only part of the information: the end date without the start date. Those examples were collected while trying to standardize a small set of diabetic cardiovascular disease trials from ClinicalTrials.Gov. To cope with a larger set of all trials within ClinicalTrials.Gov it would be necessary to first map the extent of the work needed for standardization.

The need for standardization was published in the past in [20,21]. This paper expands the discussion by providing details on the algorithms used and provides updates on the progress of the work since last publication.

2. Analyzing ClinicalTrials.Gov data

The analysis started on April 20th 2018, when all clinical trials with results were downloaded from ClinicalTrials.Gov. This amounted in 30,763 trials and this data was stored for further processing by multiple python scripts.

2.1 Data Organization

The data amounted to ~2.38GB in ~30K different XML files. Although python has good tools to read XML files, it is still cumbersome to process those files. Therefore, the first step is merging the data into more readable format. The first python script reads each XML file and converts the tree like XML structure to table like comma separated files that combine multiple trials together in a batch. Each line the generated file looks like a table that point to the trial, the XML path of the data, including parent information, and the data. This explosion of data allows easy location of data later of without traversing the entire XML tree. This processing resulted in 62 files each containing information on ~500 trials and took roughly 10.4GB data on disk. Note that the data is split into files to allow processing of the data in a machine with limited memory and possible future parallelization.

2.2 Data Indexing

Once the data is organized in a table like manner, it is possible to index all the data values by field and writing the number of occurrences of each text in the field in each organized data file. This resulted in 62 files amounting to 592MB on disk. However, some data items are associated. Specifically in interest in this work are units and those units are associated with title text that is a parent in XML structure. Eight such pairs were identified and indexed separately to associated a specific unit with a specific title and list all trials in that batch that are associated. For example, the unit "Cubic Centimeters" is associated with "Gross Tumor Volume" and appears once in the clinical trial NCT02947984. This resulted in 62 files resulting in 50.1MB.

2.3 Combining Indexes

Due to memory limitations and as preparation for parallel processing, the previous two scripts generated multiple index files each with a partial index. Since the data size was reduced to manageable size and the next steps of processing need the data unified, those indexes are merged and reorganized so that each file represents a field and stores all distinct occurrences of values, their count and associated trials for pairs of keys. This resulted in 387 files representing the fields in ClinicalTrials.Gov and in eight files representing unit associations. The combined index files take 577MB on disk. Although the entire database was analyzed and indexed, only a subset of the data affiliated with units is of interest in this paper.

2.4 Natural Language Processing of Unit Proximity

When combining only unit tags indexed files and concatenating those together there are 23,081 units, yet some of those are repeated since units appear in different fields. Disregarding repetitions, there were 21,094 unique units detected within the clinical trials with results. This number is unreasonable since many of those units represent the same unit, yet are either spelled differently or misspelled completely. The initial intention was for a machine to point towards close units. However, during this work it was realized that a computer can only do part of this job and human intervention is necessary. So the machine part was to organize the units and suggest similar units to the user. To do this, several levels of processing were required.

First, all units from ClinicalTrials.Gov were represented in several forms: 1) Using Unicode text, 2) Using ASCII text with escape characters for Unicode characters to be used as a key to avoid visual confusion with special characters and emphasize those, 3) Using translated unit text where parts of the unit were replaced by common synonym text such as "Percent" was replaced by "%" and "Year" was replaced by "yr". The latter translation was important to help the following algorithms cope with cases where spelling of units is very different, yet there is association between the units.

Additional auxiliary units were added to the database by processing CDISC units that also include synonyms and the Unified Code for Units of Measure (UCUM) [22] representations matching each unit. Each of those units was added to the list of units with registration of the CDISC unit code. Overall, 6,645 units were added by processing CDISC units.

The computer was then posed with a problem of assessing the distance/proximity between each of the ClinicalTrials.Gov units to the combined unit set that also includes CDISC units. This was done by utilizing two different distance functions in two python libraries.

The term frequency–inverse document frequency (TF-IDF) class implemented in scikit learn in TfidfVectorizer [23] was applied to the data while finding similarities with n-gram. The term n-gram in the context of this work represents a set of consecutive characters of length of 3 to 6 characters - basically sub strings of units. This TfidfVectorizer class was used to learn the inverse document frequency of the n-grams in ClinicalTrials.Gov , resulting in the matrix *A* of size (21094, 165123). The rows of the matrix represent the unique units while the columns represent the n-grams generated and the value of the matrix members represents how rare/common is the character n-grams across the different units. Then the transformation learned was used to transform CDISC units to create the matrix *B* of size (6645, 165123) that shows how how similar are the n-grams used to evaluate ClinicalTrials.Gov units are common/rare in CDISC units. Both matrices were merged to a sparse matrix *C* of size (27739, 165123) that will be used in the next stage of finding similarity score between units using a cosine transform. This is implemented as an L2 normalized dot product of the matrix *A* and *C* within sckit-learn by the cosine_similarity method [24]. The result of this operation creates a similarity matrix *D* of size (21094, 27739) that for each row representing a different unit in ClinicalTrials.Gov provides its similarity measure to other units in ClinicalTrials.Gov and in CDISC combined.

However, this similarity measure is only one possible measure of distance between words and another measure was added to compliment it using the difflib.SequenceMatcher function from the python standard library. The algorithm is supposed to implement the The Gestalt Approach to pattern matching described in [25] and initially developed by John W. Ratcliff and John A. Obershelp in 1983 - the source code of the python implementation is available in [26]. This similarity score is used to score each unit pair just like before.

The similarity of any unit is then calculated by averaging both similarity scores. For each unit from ClinicatTrials.Gov all similar units are now known by looking at the similarity matrix. The resulting matrix size is 3.31GB in size in compact binary format.

One last step is writing the units match file that contains all ClinicalTrials.Gov units and for each unit, all other units are sorted by similarity order. For unit pairs with equivalent similarity score, an importance measure is used to break the tie. Units resembling CDISC units are considered more important than units similar to other ClinicalTrials.Gov units and units that appear more times in ClinicatTrials.Gov are considered more important. The first 200 similar units are then saved to a text file along side information on unit similarity and importance. The resulting file size is 352MB and this file is the base for the next calculation step.

2.5 Unit Clustering using Machine Learning

Knowing the textual similarity between units is not sufficient for the purposes of human processing. It allows showing each unit alongside possible suggestions for synonyms. However, a human presented with one unit at a time in an order that may seem random to a human is not as efficient as showing the humans units that are close to each other together in a cluster and suggesting synonyms for each one of the units. This also allows effective distributions of the work onto groups of units for different humans to process.

To allow this, unsupervised learning using clustering algorithms was used. Specifically the MiniBatchKMeans clustering technique in scikit-learn was used [27]. This technique was chosen since it is more conservative in memory requirements from other techniques. It also allows to choose the number of clusters and 100 clusters were chosen.

The unit display order can then be determined for each unit cluster according to the distance from the cluster center. However, even with a clustering technique used, the unit clusters were not very pleasing at times. Specifically, many irregularities such as a unit that does not belong to a cluster was listed or was placed in between two similar units. To improve this situation a few methods were employed to enhance the clustering.

First, the distance function between units pairs d_{ij} , that is extracted from the similarity matrix D and is denoted as d for simplicity, was modified so each element in it is raised by a power. This way closer units are considered much more close by the clustering algorithm compared to other units that are farther away.

The clustering algorithm is repeated 3 times with 3 different distance functions for each unit pair of $d_1=d$, $d_2=d^{1.44}$, $d_3= d^{1.44*1,44} \sim d^{2}$. Each one of these clustering runs produces a different cluster number c_1, c_2, c_3 for each unit where the last clustering run favors very close units. After all clusters have been determined, each unit was assigned a combined mapping to a combined cluster defined by the vector $[c_1, c_2, c_3]$. This kind of classification served as a splintering mechanism that strongly favored close units. However, it also splintered the data into many more clusters than necessary with many clusters being very small. Figure 1 shows the splintering effect.

The combines unit cluster is *[c1,c1,c1]* c1

Figure 1. The splintering effect of multiple clustering passes

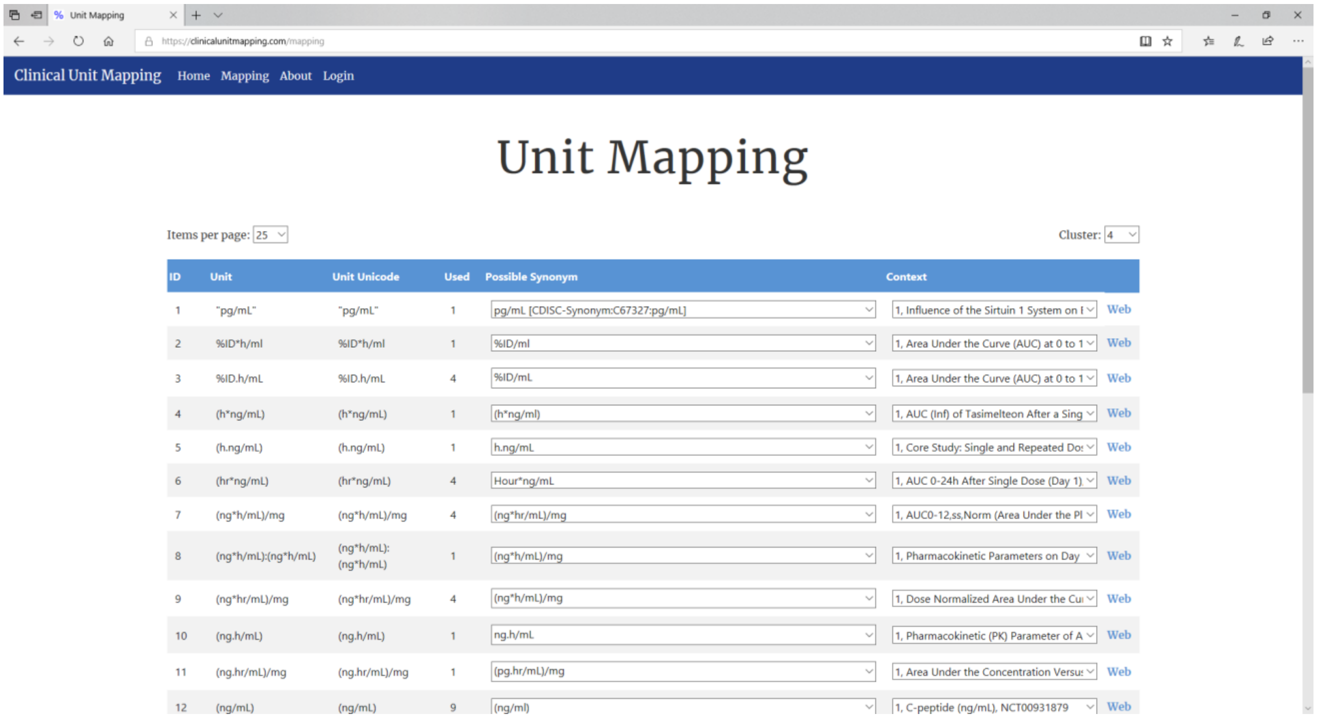
Too many clusters were not desired and therefore a last collection step was employed to gather splintered units are reattach them to the clusters. The method traversed all clusters with less than 50 units, reattaching each unit to the closest unit in a cluster larger than 50 in size. This process resulted in relocating 8,386 units to create 110 clusters. The result from this complex clustering method was pleasing enough for a human to see similarities between large batches of consecutive units that can be displayed.

2.6 Graphic User Interface

A Graphic user interface using the python PyQT library was developed as a tool to help visualize data and was presented in [20,21]. It allowed analysis and visualization of each cluster as well as the combined cluster. It allowed focusing on a unit while showing the Clinical trials it is used in and even opening the proper location in ClinicalTrials.Gov. However, it was a standalone tool that is hard to deploy. Therefore a multi user approach was selected with web deployment.

2.7 Web Portal

The web application accessible through ClinicalUnitMapping.com [28] allows users to directly interact with the unit mapping capability, including letting users map units themselves based on their own experiences. Figure 2 shows one page from a single clusters of units on the web site. The application is built in Python 3 using the Flask library as the framework for structuring the logic of how the website operates. All user account parameters and unit variables are stored in separate Sqlite3 databases, allowing the unit variables database to be upgraded without impacting user accounts. A user account is not required for viewing the mappings, but is needed for adding custom unit mappings.



Unit Mapping

Items per page: 25

Cluster: 4

ID	Unit	Unit Unicode	Used	Possible Synonym	Context	
1	"pg/mL"	"pg/mL"	1	pg/mL [CDISC-Synonym:C67327:pg/mL]	1, Influence of the Sirtuin 1 System on I	Web
2	%ID*h/ml	%ID*h/ml	1	%ID/ml	1, Area Under the Curve (AUC) at 0 to 1	Web
3	%ID.h/mL	%ID.h/mL	4	%ID/mL	1, Area Under the Curve (AUC) at 0 to 1	Web
4	(h*ng/mL)	(h*ng/mL)	1	(h*ng/ml)	1, AUC (Inf) of Tasimelteon After a Sing	Web
5	(h.ng/mL)	(h.ng/mL)	1	h.ng/mL	1, Core Study: Single and Repeated Do	Web
6	(hr*ng/mL)	(hr*ng/mL)	4	Hour*ng/mL	1, AUC 0-24h After Single Dose (Day 1),	Web
7	(ng*h/mL)/mg	(ng*h/mL)/mg	4	(ng*hr/mL)/mg	1, AUC0-12,ss,Norm (Area Under the Pl	Web
8	(ng*h/mL):(ng*h/mL)	(ng*h/mL): (ng*h/mL)	1	(ng*h/mL)/mg	1, Pharmacokinetic Parameters on Day	Web
9	(ng*hr/mL)/mg	(ng*hr/mL)/mg	4	(ng*h/mL)/mg	1, Dose Normalized Area Under the Cu	Web
10	(ng.h/mL)	(ng.h/mL)	1	ng.h/mL	1, Pharmacokinetic (PK) Parameter of A	Web
11	(ng.hr/mL)/mg	(ng.hr/mL)/mg	1	(pg.hr/mL)/mg	1, Area Under the Concentration Versu	Web
12	(ng/mL)	(ng/mL)	9	(ng/ml)	1, C-peptide (ng/mL), NCT00931879	Web

Figure 2. Snapshot of one cluster from ClinicalUnitMapping.com

All unit mappings are visualized within a cluster based table that shows each unit, the Unicode value for that unit, how many times the unit has been used in previous trials, possible synonyms for the unit, and the context the unit is used in along with a link to the trials specific ClinicalTrials.Gov page. If the visitor has a user account, a user specific mapping column appears for them to create custom mappings that only they will see. A special button allows the user to select a synonym from a drop down box and copy that value into the relevant mapping cell, saving them time. The user can save their custom mappings by clicking the save button, which will write the user's values to the database. These custom mappings will only be visible to the user who created them.

The current structure can scale based on both more units and more users. The mapping table is paginated, meaning that only a subset of database values are requested, downloaded and shown to each user at any given time. Not only does this minimize server load, but it increases website responsiveness to the end user. The database can be upgraded to PostgreSQL or MySQL for increased performance and reliability.

HTML5 and CSS3 are used for website structure and design, with JQuery 3.3.1 being used for website animations. Bootstrap 4 is used for providing many of the design components, including the navbar, buttons, forms, and pagination control. Docker containers are used for the application deployment, with a container for the application, a container for the Nginx web proxy, and a container for automating the Let's Encrypt SSL certificate operations. The complete system is hosted on a dedicated virtual private server (VPS) within one of Digital Ocean's NYC based data centers.

3. Discussion

The standardization tools described in this paper are only a few initial steps in the long journey of making medical knowledge machine comprehensible. This current effort is only possible since the last few decades focused on making medical knowledge machine readable through efforts such as scanning journal papers to be available online, Electronic Medical Records (EMR), and databases such as ClinicalTrials.Gov. This effort of standardization will improve data quality and may reveal many inconsistencies related to limits of our computational comprehension of medical data. The introduction of medical models built on standardized data already reveals gaps in our understanding of clinical trials that can be visually displayed [29]. This large gap shows how far we are from comprehension of the data. However, with standardized data, improvements in machine learning, and increases in computing power, it will be possible to reduce this comprehension gap.

It is hard to predict when this cumulative comprehension gap will become narrow enough such that it would be possible for a machine to make decisions that would be on average more accurate than a human decision.

However, when this happens, it will be possible to have applications such as an artificial medical doctor as a smart phone app that will be able to monitor human biomarkers and write prescriptions, that could be ordered online for quick delivery to the patient. However, before that happens, the medical data has to become cleaner and standardized towards computer use. Also many regulatory and human social structures will need to be challenged and changed to allow such applications. And even with such Artificial Intelligence (AI) applications in place, the role of the human in the loop may be important as can be seen from a past example of computer chess [30] where a computer has reached dominance in the game within a century, yet a combination of human and computer in chess is still considered powerful. So even before a medical AI is available, it may be possible to aid human doctors with computer applications. Although this work is aimed eventually at computer comprehension, having standardized units will help humans in the short term by visualizing differences between clinical trials numerical results side-by-side. Moreover, standardization of ClinicalTrials.Gov data will help other visualization applications such as those described in [29, 31-33].

4. Conflict of Interest Declaration

The work reported here was self funded and was not endorsed by any external entity and the direct product of this work - medical unit standard is not intended to be commercialized. The current intention is to contribute it to CDISC through SISO to be eventually merged into UMLS. Proposals to support this effort are pending and not yet approved.

Note that Jacob Barhak holds one patent on disease modeling technology [34] and has another pending patent that has claims on unit conversion [35]. Although the work presented here may eventually enable use of those technologies, there is currently no overlap and a unit standard potentially produced based on this work will not constitute conflict of interest, nevertheless interests are declared.

5. Reproducibility Information

At the time of writing those words the paper can be reproduced from the following sources, the processing reported in this paper is archived in the file AnalyzeCT_GOV_Code_2019_01_16.zip. The data from CinicalTrials.Gov used in processing is archived in StudiesWithResults_Downloaded_2018_04_20.zip . The web site database for the beta version is in the file: PartUnitsDB_2018_12_26.db . The web site source code is stored a private repository in Github: camisatx/clinical-trials Timestamp: Jan 15, 2019 11:50:31 -0600.

6. 7. References

[1] Ryan F. Flanagan, Olaf Dammann: "The Epistemological Weight of Randomized-Controlled Trials Depends on Their Results." Perspectives in Biology and Medicine, vol. 61 no. 2, 2018, pp. 157-173. Project MUSE, https://dx.doi.org/10.1353/pbm.2018.0034

[2] Stephen Garba, Adamu Ahmed, Ahmed Mai, Geoffery Makama, and Vincent Odigie: "Proliferations of Scientific Medical Journals: A Burden or A Blessing" . Oman Med J. 2010 Oct; 25(4): 311–314. https://dx.doi.org/10.1353/pbm.2018.003410.5001/omj.2010.89 , PMCID: PMC3191655 , PMID: 22043365

[3] Adam Miller: "The future of health care could be elementary with Watson". CMAJ. Jun 11, 2013; 185(9): E367–E368. https://dx.doi.org/10.1503%2Fcmaj.109-4442 . Also available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3680569

[4] Watson and the Jeopardy! Challenge. YouTube Online: https://www.youtube.com/watch?v=P18EdAKuC1U

[5] Self-Driving Car, Wikipedia, https://en.wikipedia.org/wiki/Self-driving_car

[6] WHO, Classifications, ICD 11 is here - https://www.who.int/classifications/icd/en/

[7] SNOWMED CT The Global Language of Healthcare: Online: https://www.snomed.org/snomed-ct

[8] NLM, Overview of SNOWMED CT, Online: https://www.nlm.nih.gov/healthit/snomedct/snomed_overview.html

[9] Fast Healthcare Interoperability Resources (FHIR) - https://www.hl7.org/fhir/

[10] CDISC strength through collaboration, Online: https://www.cdisc.org/

[11] Unified Medical Language System (UMLS) , Online: https://www.nlm.nih.gov/research/umls/

[12] The Reference Model for Diease Progression, SimTK, https://simtk.org/projects/therefmodel

[13] Jacob Barhak: "The Reference Model: A Decade of Healthcare Predictive Analytics with Python", PyTexas 2017, Nov18-19, 2017, Galvanize, Austin TX. Presentation: http://sites.google.com/site/jacobbarhak/home/PyTexas2017_Upload_2017_11_18.pptx Video: https://youtu.be/Pj_N4izLmsI

[14] Deborah A. Zarin, Tony Tse, Rebecca J. Williams, Sarah Carr: "Trial Reporting in ClinicalTrials.gov — The Final Rule". N Engl J Med ; 375 pp. 1998-2004, 2016 http://dx.doi.org/10.1056/NEJMsr1611785

[15] Nicholas C. Ide, Russell F. Loane, Dina Demner-Fushman: "Essie: A concept-based search engine for structured biomedical text". J Am Med Inform Assoc. 14(3) pp. 253-63, 2007 https://doi.org/10.1197/jamia.M2233

[16] ClinicalTrials.Gov Trends, Charts, and Maps – Online: https://clinicaltrials.gov/ct2/resources/trends

[17] PUBLIC LAW 110–85—SEPT. 27, 2007 - TITLE VIII—CLINICAL TRIAL DATABASES . Section 801 of the Food and Drug Administration Amendments Act of 2007. Online: https://www.gpo.gov/fdsys/pkg/PLAW-110publ85/pdf/PLAW-110publ85.pdf#page=82

[18] Jacob Barhak: "The Reference Model for Disease Progression Combines Disease Models". I/IITSEC 2016 28 Nov – 2 Dec 2016, Orlando Florida. Paper: http://www.iitsecdocs.com/volumes/2016

[19] Jacob Barhak: "The Reference Model Models ClinicalTrials.Gov". SummerSim 2017 July 9-12, Bellevue, WA. Paper: https://doi.org/10.22360/SummerSim.2017.SCSC.022

[20] Jacob Barhak, Chris Myers, Leandro Watanabe, Lucian Smith, Maciek Jacek Swat: "Healthcare Data and Models Need Standards". Simulation Interchangeability Standards Organization (SISO) 2018 Fall Innovation Workshop". 9-14 Sep 2018 Orlando, Florida. Presentation: http://sites.google.com/site/jacobbarhak/home/SISO_SIW_2018_08_14.pptx

[21] Jacob Barhak: "Python Based Standardization Tools for ClinicalTrials.Gov". Combine 2018 . Boston University. October 8-12, 2018. Poster: http://co.mbine.org/system/files/COMBINE_2018_Barhak.pdf

[22] The Unified Code for Units of Measure (UCUM), Online: http://unitsofmeasure.org/

[23] Scikit-learn: TfidfVectorizer class documentation: Online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[24] Scikit-learn: cosine similarity method documentation. Online: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

[25] John W. Ratcliff, David E. Metzener: "Pattern Matching: The Gestalt Approach", Online: https://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/DDJ/1988/8807/8807c/8807c.htm

[26] Module difflib Online: https://svn.python.org/projects/python/trunk/Lib/difflib.py

[27] Scikit-learn: cluster.MiniBatchKMeans documentation. Online https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html

[28] Joshua Schertz , Jacob Barhak: "Clinical Unit Mapping" - online: clinicalunitmapping.com

[29] Jacob Barhak: "The Reference Model Visualizes Gaps in Computational Understanding of Clinical Trials", 2018 IMAG Futures Meeting March 21-22, 2018 @ NIH, Bethesda, MD. http://sites.google.com/site/jacobbarhak/home/Poster_IMAG_MSM2018_Map_Upload_2018_03_17.pdf

[30] Computer chess, Wikipedia, https://en.wikipedia.org/wiki/Computer_chess

[31] Spencer Phillips Hey, Jessica M. Franklin, Jerry Avorn, and Aaron S. Kesselheim: "Success, Failure, and Transparency in Biomarker-Based Drug Development A Case Study of Cholesteryl Ester Transfer Protein Inhibitors". Circulation: Cardiovascular Quality and Outcomes. 2017;10:e003121 , https://doi.org/10.1161/CIRCOUTCOMES.116.003121

[32] Spencer Phillips Hey, Charles M Heilig, Charles Weijer: “Accumulating Evidence and Research Organization (AERO) model: a new tool for representing, analyzing, and planning a translational research program” Trials 2013 14:159, https://doi.org/10.1186/1745-6215-14-159

[33] PORTAL Biomarker Research Consortium, Graphing a Biomarker-Driven Research Program, Online: https://www.portalresearch.org/aero-graph.html

[34] Jacob Barhak: "Reference model for disease progression". United States Patent 9,858,390, January 2, 2018

[35] Jacob Barhak: "Analysis and Verification of Models Derived from Clinical Trials Data Extracted from a Database". US patent Utility application #15466535

Author Biographies

JACOB BARHAK specializes in population modeling and specifically in chronic disease modeling with emphasis on using computational technological solutions. Dr. Barhak has diverse international background in engineering and computing science. The Reference Model for disease progression was independently self-developed by Dr. Barhak in 2012. He is the developer of the MIcro Simulation Tool (MIST). See: http://sites.google.com/site/jacobbarhak/

JOSHUA SCHERTZ is a freelance programmer with background in energy, finance, and aerospace. He is an experienced Python programmer and worked on over a dozen programming projects many of which can be accessed through his web site: https://joshschertz.com/