Standardization of Tumor Names in NIH-Clinical Trials Registry using Large Language Model Embedding Analysis

Aditya Lahiri1, Sangeeta Shukla1, Ben Stear1, Taha Mohseni Ahooyi1, Deanne Taylor1,2

**Affiliations:**

1. The Department of Biomedical and Health Informatics, The Children’s Hospital of Philadelphia, Philadelphia PA
2. Department of Pediatrics, University of Pennsylvania Perelman Medical School, Philadelphia PA

**Key Words:** Clinical Trials Registry, WHO Tumor Classification, Text Embedding, Clinical Text Standardization

**Word Count:**

**ABSTRACT**

**Objective:**

This project aimed to extract tumor names from the National Institute of Health's (NIH) Clinical Trials Registry (ClinicalTrials.gov) and standardize them according to the corresponding tumor terminology established in the World Health Organization's (WHO) Tumours Classification System.

**Materials and Methods:**

We developed a computational pipeline that loads the disease data file from NIH's clinical trials registry and identifies tumors from the rest of the diseases. Following the tumor identification, each tumor from the registry is mapped to the closest matching terminology in the WHO Tumor Classification System using twelve text standardization methods based on text-similarity, text-embedding, and clustering methods. We evaluate each of these methods on a subset of tumors derived from the registry to evaluate their accuracies in mapping the tumors to their standardized tumor terminology in the WHO Tumours Classification System.

**Results:**

Our results indicated that embedding-based text standardization outperformed methods based on text-matching algorithms and clustering. We also noticed that the accuracies of clustering-based methods improved significantly when text embeddings were used to measure divergence in the clustering algorithm instead of text-similarity metrics. Overall, we found that finding the closest WHO term to a given tumor name from the registry using Euclidean distance outperformed the other methods.

**Discussion and Conclusion:**

The tumor names in the NIH Clinical Trials Registry are not standardized, making integrating this data with other databases challenging. Therefore, we developed a computational pipeline that identifies tumors from the NIH Clinical Trials Registry and standardizes them according to the standardized terms established in the WHO Tumours Classification System.

**INTRODUCTION**

Cancer is a major global health problem [[1]](https://paperpile.com/c/NPPxEM/Weij) and is the second largest cause of deaths in the United States [[2]](https://paperpile.com/c/NPPxEM/6kRx). In the United States it is estimated that there will be over 2 million cases of newly diagnosed cancer and 611,720 deaths in 2024[[2]](https://paperpile.com/c/NPPxEM/6kRx). Among children (ages 0 to 14 years) and adolescents (ages 15 to 19 years) , in the US, pediatric cancer persists to be the second and fourth leading cause of deaths[[2]](https://paperpile.com/c/NPPxEM/6kRx), despite the jump in 5-year survival rate to 80% in the last 5 decades [[3]](https://paperpile.com/c/NPPxEM/tUKD) [[4]](https://paperpile.com/c/NPPxEM/FRbF). The increased survival rate of pediatric cancer treatment can primarily be attributed to successes in the treatment of common childhood hematological malignancies such as acute lymphoblastic leukemia, where the survival rate has risen from 10% to 90%[[5]](https://paperpile.com/c/NPPxEM/68hp), thus the success is not uniformly shared across all types of pediatric cancers [[6]](https://paperpile.com/c/NPPxEM/NRZI) [[6]](https://paperpile.com/c/NPPxEM/NRZI)especially cancers of the brain and nervous system[[2]](https://paperpile.com/c/NPPxEM/6kRx). Compared to adult cancers, pediatric cancers are rarer and with fewer available therapeutic agents that have been tested in clinical trials due to challenges associated with recruiting statistically significant and diverse pediatric populations to support the various phases of clinical trials, logistical issues related to clinical trial-site location and molecular heterogeneity of tumors[[7,8]](https://paperpile.com/c/NPPxEM/Lq9U+5cZt).

The Food and Drug Administration Modernization Act of 1997 (FDAMA) mandated the National Institutes of Health (NIH) to create a publicly available resource to disseminate information on the effectiveness of drugs in federally or privately funded clinical trials (CT) [[9]](https://paperpile.com/c/NPPxEM/hYaH). This eventually led to the creation of the ClinicalTrials.gov registry, which was publicly launched on February 29, 2000. Since then, the registry has amassed over 482,529 research studies across all 50 states in the US and 223 countries [[9]](https://paperpile.com/c/NPPxEM/hYaH). Each record within the CT registry is self-reported by the trial sponsor through the web-based data entry platform known as the Protocol Registration and Results System (PRS) [[9]](https://paperpile.com/c/NPPxEM/hYaH). The registry requires sponsors to enter basic details regarding the trial, such as purpose, design, patient eligibility criteria, and other critical information about the study [[10]](https://paperpile.com/c/NPPxEM/Hrsy). The CT registry requires by law that data be entered in a tabular format and that an individual with knowledge of study design and data analysis be involved in the submission process to ensure that results are appropriately summarized and the data submission is consistent with the review criteria of the CT registry. Following the submission of a record, the record is reviewed internally by CT registry staff before it is posted publicly on ClinicalTrials.gov.

Even with the established protocols and guidelines in the submission process, the registry data may contain various inconsistencies in the form of extraneous information, typographical errors, missing values, etc. Such discrepancies must be addressed or filtered out before the data can be used for further downstream analysis. In this study, we developed a computational pipeline to standardize the tumor names contained in the "conditions" data file in the CT registry. The conditions data includes the names of the diseases or conditions that are the subject of the trial. Among the various diseases present in the CT registry, we focused on cancers as they are a leading cause of death in the US and the world [[11]](https://paperpile.com/c/NPPxEM/GzPE). Thus, standardizing cancer names in the CT registry , i.e. by mapping each tumor name in the CT registry to their equivalent standardized names in the World Health Organization's tumor classification system or the National Cancer Institute terms (NCIT) will allow us to integrate tumor data from the CT registry with other databases such as Open Targets or Illuminating the Druggable Genome will allow us to deeply understand the landscape of tumors, targets, and drugs.

It must be noted that the CT registry recommends adding relevant Medical Subject Headings (MeSH) terms or terms from another controlled vocabulary, such as the Systematized Nomenclature of Medicine—Clinical Terms (SNOMED CT) , that has been mapped to MeSH within the Unified Medical Language System (UMLS) metathesaurus for each of the conditions \cite{CT.gov}. While this recommendation adds a level of standardization to the disease/condition names present in CT registry, the MeSH terms by themselves often fall short of describing the disease or condition accurately. Furthermore, for a given disease/condition there may be multiple associated MeSH terms, thereby leaving it to the user of the data to establish the most appropriate MeSH term for that disease/condition. Additionally, for certain records in the CT registry, there might be no associated MeSH terms provided for a disease/condition name, therefore solely using the MeSH terms for analyzing the diseases or conditions that are the subject of a clinical trial record is not reliable. In table 1, we show examples of CT records with their disease/condition names and associated mesh terms. Table 1 was created by performing a full join on the files “conditions.txt” and “browse\_conditions.txt”, the files were joined on CT ID.

**Table 1: Conditions data with MeSH Terms**

| CT ID | Condition name | MeSH term |  |
| --- | --- | --- | --- |
| NCT05082610 | triple negative breast cancer | neoplasms,triple negative breast neoplasms,carcinoma, non-small-cell lung,breast neoplasms,neoplasms by site,breast diseases,skin diseases,carcinoma, bronchogenic,bronchial neoplasms,lung neoplasms,respiratory tract neoplasms,thoracic neoplasms,lung diseases,respiratory tract diseases |  |
| NCT04254107 | triple negative breast cancer | lymphoma,carcinoma,lymphoma, t-cell, peripheral,lymphoma, large b-cell, diffuse,triple negative breast neoplasms,squamous cell carcinoma of head and neck,stomach neoplasms,neoplasms by histologic type,neoplasms,lymphoproliferative disorders,lymphatic diseases,immunoproliferative disorders,immune system diseases,neoplasms, glandular and epithelial,neoplasms by site,carcinoma, squamous cell,lymphoma, b-cell,lymphoma, non-hodgkin,lymphoma, t-cell,breast neoplasms,breast diseases,skin diseases,head and neck neoplasms,gastrointestinal neoplasms,digestive system neoplasms,digestive system diseases,gastrointestinal diseases,stomach diseases |  |
| NCT01590680 | neuroblastoma | neuroblastoma,pheochromocytoma,paraganglioma,neuroectodermal tumors, primitive, peripheral,neuroectodermal tumors, primitive,neoplasms, neuroepithelial,neuroectodermal tumors,neoplasms, germ cell and embryonal,neoplasms by histologic type,neoplasms,neoplasms, glandular and epithelial,neoplasms, nerve tissue,neuroendocrine tumors |  |
| NCT04081701 | medulloblastoma | adenoma,meningioma,medulloblastoma,paraganglioma,pituitary neoplasms,esthesioneuroblastoma, olfactory,central nervous system neoplasms,hemangioblastoma,neoplasms, glandular and epithelial,neoplasms by histologic type,neoplasms,pituitary diseases,hypothalamic diseases,brain diseases,central nervous system diseases,nervous system diseases,endocrine system diseases,neoplasms, nerve tissue,neoplasms, vascular tissue,meningeal neoplasms,nervous system neoplasms,neoplasms by site,glioma,neoplasms, neuroepithelial,neuroectodermal tumors,neoplasms, germ cell and embryonal,neuroectodermal tumors, primitive,neuroendocrine tumors,endocrine gland neoplasms,hypothalamic neoplasms,supratentorial neoplasms,brain neoplasms,neuroblastoma,neuroectodermal tumors, primitive, peripheral,olfactory nerve diseases,cranial nerve diseases,hemangioma, capillary,hemangioma |  |
| NCT04294784 | gastroesophageal cancer | NA |  |
| NCT02669914 | gastroesophageal cancer | lung neoplasms,carcinoma, non-small-cell lung,colorectal neoplasms,pancreatic neoplasms,ovarian neoplasms,brain neoplasms,kidney neoplasms,carcinoma, renal cell,breast neoplasms,respiratory tract neoplasms,thoracic neoplasms,neoplasms by site,neoplasms,lung diseases,respiratory tract diseases,carcinoma, bronchogenic,bronchial neoplasms,intestinal neoplasms,gastrointestinal neoplasms,digestive system neoplasms,digestive system diseases,gastrointestinal diseases,colonic diseases,intestinal diseases,rectal diseases,endocrine gland neoplasms,pancreatic diseases,endocrine system diseases,ovarian diseases,adnexal diseases,genital diseases, female,female urogenital diseases,female urogenital diseases and pregnancy complications,urogenital diseases,genital neoplasms, female,urogenital neoplasms,genital diseases,gonadal disorders,central nervous system neoplasms,nervous system neoplasms,brain diseases,central nervous system diseases,nervous system diseases,urologic neoplasms,kidney diseases,urologic diseases,male urogenital diseases,adenocarcinoma,carcinoma,neoplasms, glandular and epithelial,neoplasms by histologic type,breast diseases,skin diseases |  |

In Table 1, for the condition triple negative breast cancer with CT ID: NCT05082610, the most appropriate MeSH terms is “triple negative breast neoplasms”, however, there are other associated MeSH terms such as “carcinoma, non-small-cell lung” and “respiratory tract diseases” which do not describe the condition of triple negative breast cancer. Since there is no metric in the CT registry by which we can computationally determine the most appropriate MeSH terms for a given condition. Furthermore, the MeSH terms are not identical between studies where the condition names are the same, which adds to the inconsistencies between records with same condition names. Consider studies with CT ID: NCT05082610 and NCT04254107, in both the studies the condition names are “triple negative breast cancer”, however, the MeSH terms are not identical. For instance, in the study with CT ID: 04254107 contains various MeSH terms associated with lymphomas such as “lymphoma, b-cell” , “large b-cell”, “lymphoma, large b-cell, diffuse”, “lymphoma, t-cell, peripheral” etc, these terms are not contained in the study with CT ID: NCT05082610. We can also see in study NCT04294784 , for the condition “gastroesophageal cancer” , there are no MeSH terms , however, for the same condition with a different CT ID NCT02669914 has multiple associated MeSH terms. Due to these inconsistencies, and a lack of metric to determine the most accurate MeSH term for a given condition, we can determine that using the MeSH terms to describe the conditions is not reliable. Therefore, we decided to use the conditions filed in the CT registry to extract the disease/conditions that are the subject of a CT. This study focuses on extracting tumors from the conditions field in the CT registry and standardizing them by matching them to their closest matching terms in the WHO tumor classification system or NCIT.

**METHODS**

**Data Availability**

The data used in this paper is obtained from the NIH-Clinical Trials registry (<https://clinicaltrials.gov/>)[[9]](https://paperpile.com/c/NPPxEM/hYaH). The data can be publicly accessed via the Clinical Trials API or from the Aggregate Analysis of ClinicalTrials.gov-Clinical Trials Transformative Initiative (AACT-CTTI) website (<https://aact.ctti-clinicaltrials.org/download>). ​​The ACCT-CTTI website is updated daily with contents from ClinicalTrials.gov and a static database is made available at the start of each month. The static database contains information about all the studies registered in ClinicalTrials.gov. We downloaded a copy of the database from the ACCT-CTTI website on August 22, 2023 which is available at the following at the ACCT-CTTI as a pipe-delimited file under the name of “20230822\_export.zip” .

**Data Extraction Pipeline**

The clinical trials database contains information about various aspects of the study, such as outcomes, interventions used, conditions (diseases) studied, design of experiments, sponsors of the studies, etc. This information is presented to the public in individual text files. Each text file contains the National Clinical Trial Identification Number (NCTID), which serves as the unique identifier (foreign key) for that study. The NCTID allows one to reference a particular clinical trial and aggregate various information associated with it, which is stored across all the text files in the database. In this study, we select the conditions ("conditions.txt") and interventions ("intervention.txt") files, which contain information regarding diseases and drugs used in each of the clinical trial studies, respectively. The conditions file contains 801,197 records and is annotated with the following fields: "*id*," "*nct\_id*," "*name*," and "*downcase\_name*." The "id" field represents the identification number for that record within the conditions file, whereas the "*nct\_id*" is the foreign key for that record. The "*name*" and "*downcase\_name*" fields contain names of the diseases studied in a specific clinical trial study, with the "downcase\_name" containing the disease name in "name" in downcase format. We identified 105483 unique diseases (by uniqueness of strings) in the conditions file.

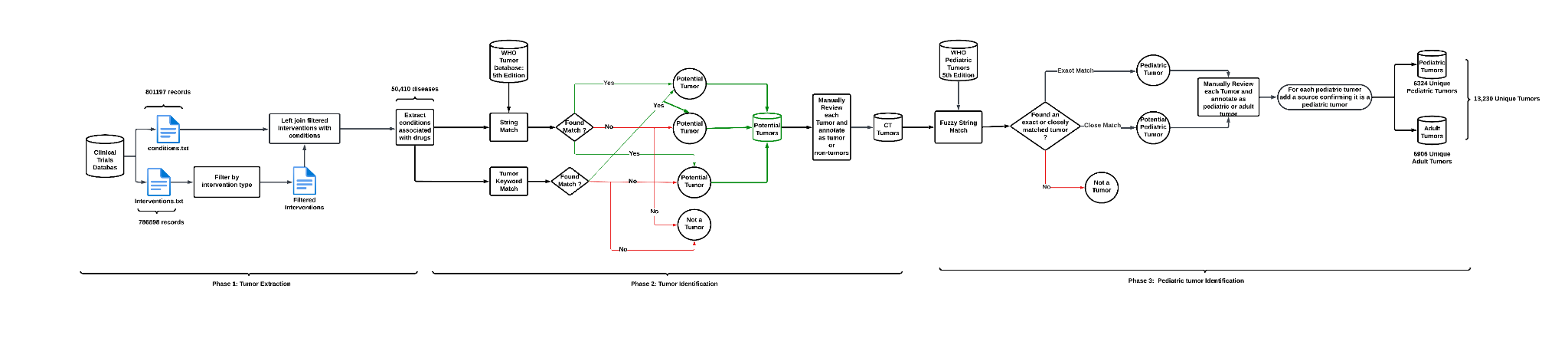
The conditions file does not classify the disease based on any system, such as the WHO's International Classification of Diseases (ICD). Therefore, we needed to develop a computational pipeline to extract tumor names from the rest of the diseases. In this work, we focused on extracting tumors with an associated drug target. To achieve this, we needed to subset the diseases in the conditions file (conditions.txt) with the types of drugs registered in the interventions (interventions.txt) file. The intervention file has 786,898 records and is annotated with the following fields: "*id*," "*nct\_id*," "*intervention\_type*", "*name*," and "*description*." The "id" field represents the identification number for that record within the conditions file, whereas the "*nct\_id*" is the foreign key for that record. The "*name*" and "*description*" fields provide names and details regarding the interventions used in that particular clinical trial, and the "*intervention\_type*" field classifies the interventions into one of the eleven distinct intervention types listed in Table 2.

**Table 2: Types of interventions listed in interventions file in the NIH-Clinical Trials Registry**

| Index | Type of intervention | Included in analysis |
| --- | --- | --- |
| 1 | Drug | Yes |
| 2 | Biological | Yes |
| 3 | Radiation | No |
| 4 | Device | No |
| 5 | Behavioral | No |
| 6 | Other | No |
| 7 | Genetic | Yes |
| 8 | Procedure | No |
| 9 | Combination Product | Yes |
| 10 | Dietary Supplement | No |
| 11 | Diagnostic Test | No |

For extracting tumors with an associated drug target, we first join the intervention file with the conditions file using the foreign key NCTID ("nct\_id") and then filter the file for interventions belonging to the following type: "*Drug*", "*Biological*", "*Combination Product*", and "*Genetic*" in The rationale behind limiting our diseases to these intervention types was to ensure that there is a corresponding targeted or chemotherapy, immunotherapy-based treatment option for the tumors from the clinical trials database. After filtering based on the intervention types, we obtained 50,410 unique diseases from which we needed to identify the tumors. The tumor extraction pipeline process is further detailed as phase 1 of the of the tumor extraction and annotation pipeline in figure 1.

**Figure 1**: **Tumor extraction and annotation pipeline**

’

The second phase of this pipeline aimed at differentiating tumors from the rest of the diseases. The pipeline achieves this by employing two independent protocols to detect tumors from disease names. The first protocol consisted of checking if each disease name contained a tumor key word. The tumor key words are listed below:

"cancer|carcinoma|adenocarcinoma|tumor|lymphoma|blast|myeloma|melanoma|leukemia|astrocytoma|malignant|neoplasm|neoplasia|mesothelioma|ependymoma|glioma|thymoma|waldenstrom macroglobulinemia|myelodysplastic syndrome|polycythemia vera|myelofibrosis

|myeloproliferative|sarcoma|gist-plus syndrome|macroglobulinemia|mycosis fungoides|sezary's disease|plasmacytoma"

If the disease name contained one of the above tumor key words, we flagged that disease as a potential tumor. The second protocol in detecting tumors, matched each of the diseases in the clinical trials database to the tumor names listed in the 5th edition of the WHO tumor database using a fuzzy string match algorithm. If a disease from clinical trials exactly matched a term in the WHO database, it was flagged as tumor. If the disease did not match to any tumor within the WHO database, we performed a fuzzy (approximate) match with the disease name with each term in the WHO database. This was done by computing the generalized Levenshtein edit distance of the clinical trial disease to each WHO database term. The fuzzy matching using generalized Levenshtein edit distance was implemented using the agrepl function in R [[12]](https://paperpile.com/c/NPPxEM/rIfO) and the maximum distance was set to 0.2. If there was at least one WHO Term within this maximum distance , that clinical trial disease was flagged as a potential tumor.Once a clinical trials disease was flagged as a potential tumor by either of these two protocols, they were manually validated to confirm if they were indeed a tumor. This stage of the pipeline is represented as phase 2 in figure 1.

After the manual validation step in phase 2, we identified a total of 13,230 unique (by string uniqueness) tumors in the clinical trial database. Out of these tumors, we also wanted to identify which of the tumors were pediatric and we achieved this by implementing a similar fuzzy string match algorithm as in phase 2 , however, with the only difference being that we compare the 13,230 tumors to the pediatric tumors listed in the 5th edition of the WHO tumor database instead of the entire WHO tumor database. Once the tumors are flagged as pediatric or adult by the pipeline, we manually validate the results and if the tumors are indeed pediatric we add a citation where it states that the tumor in question is a pediatric tumor. This is represented as phase 3 in the tumor extraction and annotation pipeline in figure1. All in all of the 13,230 tumors the pipeline identified 6324 to be pediatric tumors.

**Anomalies in Clinical Trials Tumor Names**

A cursory analysis of the tumor names data revealed various sources of discrepancy such as typographical errors, extraneous information, missing values, drug names entered instead of disease names, multiple tumor names etc. Furthermore, we also noticed many tumor names in the disease file did not follow standardized tumor names from the WHO Tumor Classification database (https://tumourclassification.iarc.who.int/welcome/) or the National Cancer Institute Thesaurus (NCIT) (https://ncithesaurus.nci.nih.gov/ncitbrowser/). With the tumor names not being standardized, it becomes challenging to relate the tumors in the clinical trials database to tumors in other databases, such as Illuminating the Druggable Genome (IDG) or Open Targets (OT). These discrepancies also prevented us from annotating these tumor names as pediatric or adult cancers, and consequently 144 clinical trials tumors were annotated as “DA” (Do not Annotate) in the the field designating them as pediatric tumor (“PedCanTumor) in the data file “tumor\_annotated\_adult\_ped.csv”. Table 3 outlines some of these common discrepancies associated with tumor names.

**Table 3: Discrepancies associated with Conditions Data**

| **NCT ID** | **Disease Name** | **Issue** | **Standardized WHO Tumor Name** |
| --- | --- | --- | --- |
| NCT02172768 | acute myeloid leucaemia | Leukemia is not spelled correctly. | acute myeloid leukaemia |
| NCT02658838 | lovastatin/ticagrelor [va drug interaction] | Drug names are presented instead of diseases. |  |
| NCT01291602 | healthy male and female japanese volunteers | Description of study participant is provided instead of diseases. |  |
| NCT04323956 | ann arbor stage iii follicular lymphoma | Extra information provided on staging of the tumor. | follicular lymphoma |
| NCT02637531 | adrenocortical carcinoma (part g) | Extra information presented inside parenthesis. | adrenal cortical carcinoma |
| NCT03712605 | pathologic stage i merkel cell carcinoma ajcc v8 | Extra information on staging of the tumor. | merkel cell carcinoma |
| NCT01780740 | disorder; heart, functional, postoperative, cardiac surgery | Multiple clinical terms entered in a comma separated format. |  |
| NCT01963481 | metastatic beast cancer | Breast cancer is spelled incorrectly. | invasive breast carcinoma of no special type (5th Edition WHO) ,  invasive breast carcinoma (3rd Edition WHO) |
| NCT01782235 | primary sj√∂gren's syndrome (pss) | Special characters present in disease name. | NA |
| NCT01209195 | locally advanced/metastatic or recurrent ovarian cancer, fallopian tube cancer, | Multiple tumor names |  |
| NCT01863108 | tumor vaccines | Description of tumor intervention |  |
| NCT02018874 | solid tumors and non-hodgkin's lymphoma | Solid tumors is a vague term whereas non-Hodgkin’s lymphoma is specific and a childhood tumor. |  |
| NCT05050630 | at least one positive lesion according to the 2014 lugano criteria for hodgkin's and non-hodgkin's lymphoma | Vague tumor description. It is not clear whether the tumor is Hodkin’s or non-Hodgkin’s tumor. |  |

Once the clinical trials tumor names are standardized, they can be related to external databases to draw further insights about the tumors, such as potential drug targets and currently available FDA-approved drugs.

Due to various sources of discrepancies in the tumor names in the clinical trials database, there is a need to standardize them. To standardize the tumor names, we designed a computational pipeline that takes the tumor names from clinical trials as input and annotates them according to the WHO tumor classification database, considered the gold standard. However, the WHO tumor classification database has been updated over the years and there are multiple versions of this database. Thus, we considered the 5th, 4th, and 3rd editions of the WHO tumor classification database as a reference for standardization. Furthermore, we also considered the NCIT database for standardizing the tumor names. Therefore, we designed our pipeline to generate three separate standardizations for the clinical trials tumor names with respect to the 5th edition (latest) of WHO tumors, all editions (5th, 4th, and 3rd) editions of WHO tumors, and the NCIT tumors. In the following section we will discuss the various methods used to standardize the clinical trials tumor names.

**Standardization Pipeline**

We considered various methods to standardize the names of tumors in clinical trials. These methods are based either on text-matching techniques or text-embedding-based analysis. In total, we implemented 12 different standardization techniques, of which six were based on text-matching techniques and the other six were based on embedding text-embedding-based analysis. In the following subsections, we will first discuss the text-matching techniques and then the text-embedding-based analysis.

***Text Matching Technique: Edit Distance based Matching***

When standardizing the tumor names from clinical trials to their WHO or NCIT equivalent standardized name, we are essentially comparing two pieces of text or strings, and ideally, we would want to assign the standardized tumor term that conveys the same meaning as the tumor name in the clinical trials database. Edit distances offer a way to compare the similarity between two strings. It is a metric that can be based on the minimum number of edit-operations (deletions, substitutions, insertions, etc), qgrams or heuristic that are required to transform one string into another. The larger the edit distance between two strings, the further apart the strings are; thus, two strings with minimal edit distance could potentially convey the same meaning. There are several methods to compute edit distances between strings; in this paper, we use the following three commonly used edit distances: normalized Levenshtein distance, Jarro-Winkler distance, and cosine distance. An example of how edit distances can be used to compare strings is discussed in Appendix A1. Following are brief descriptions of each method.

*Normalized Levenshtein distance*: Levenshtein distance between two strings is defined as the minimum number of single character edits which include insertions, deletions or substitutions required to transform a string to its target string. The Levenshtein distance between two strings is not normalized , thus to normalize them, we divide the Levenshtein distance by the length of the longest string, so that we get a distance in the interval [0,1]. By normalizing the Levenshtein distance, we can compare the dissimilarity between a string and multiple target strings on the same scale ([0,1]). Furthermore, we can also define the similarity between two strings as follows:

In the above equation , |s1| and |s2| represent the respective lengths of strings s1 and s2 between which we are comparing the Levenshtein similarity. We calculate Levenshtein distance using the stringdist library in the R programming language [[13]](https://paperpile.com/c/NPPxEM/MKyW). Following the calculation of the Levenshtein distance, we compute the normalizing factor (i.e. divide the Levenshtein by the longest string size) for distance between each pair of strings and normalize the Levenshtein distance.

*Jarro-Winkler distance*: The Jarro-Winkler distance is a normalized edit distance between two strings. It is a variant of the Jarro similarity measure which is defined as follows between two strings S\_1 and S\_2 respectively:

Where s1 and s2 are lengths of the strings S\_1 and S\_2 respectively , m is the number of matching characters and t is the number of transpositions. It should be noted which estimating m that two characters from S\_1 and S\_2 are only considered to be matching if they are the same and are less than characters apart.

The Jarro-Winkler similarity measure builds on top of the Jarro similarity measure and introduces two more parameters for rewards and favorable scales the Jarro similarity score if the two strings share similar prefixes. The Jarro-Winkler similarity is defined as follows:

Where *l* is defined as the length of the common prefix at the start of the string (maximum of 4 characters) , whereas *p* is a scaling factor that rewards the score for having common prefixes. Typically *p* is set to 0.1 and should not exceed 0.25 (or ¼ as the maximum length of prefix being considered is 4).

With the above definition of Jarro-Winkler similarity in place, the Jarro-Winkler distance is simply defined as follows:

We calculate the Jarro-Winkler distance using the stringdist package in the R-programming language [[13]](https://paperpile.com/c/NPPxEM/MKyW).

*Cosine Distance***:** In order to define cosine distance , we first need to define cosine similarity. For two non-zero vector vectors, cosine similarity is defined as the dot product of the two vectors divided by the product of their lengths. Cosine similarity ranges from [-1,1], with -1 representing total opposition, 0 representing complete dissimilarity, and 1 representing full similarity between the vectors. Cosine similarity between two vectors A and B is defined as follows:

However, to use cosine similarity in the context of strings, the vectors A and B represent the frequencies of unique words in strings S1 and S2. Since frequencies cannot be negative, the cosine similarity ranges between [0,1]. Thus, there is no need to normalize this metric, and cosine distance is defined simply as

We calculate the cosine distance using the stringdist package in the R-programming language [[13]](https://paperpile.com/c/NPPxEM/MKyW).

Based on the three edit distances, we compute the pairwise distances between each tumor name identified in the clinical trials database and the standardized tumor terms in each WHO database and NCIT edition. For each clinical trial tumor name, we select the nearest standardized terms under each edit distance. If more than one term qualified as the closest term, we reported them all by separating the terms with a semicolon. We report the standardized terms for each standardization reference, i.e., the 5th edition of WHO tumors, all editions (3rd, 4th, and 5th) of WHO tumors, and NCIT tumors.

***Text Matching Technique: Edit Distance combined with Affinity Propagation Clustering***

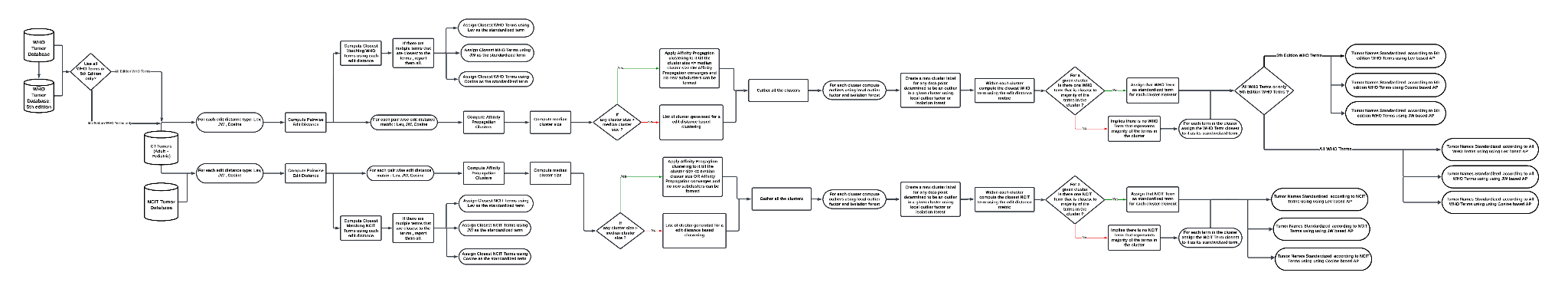
We explored another method based on edit distances that standardized the tumor names from the clinical trials database. This method first implements a clustering step, and then a standardization step is applied to each of the clusters that are formed. We used affinity propagation (AP) clustering to perform the clustering as it automatically determines the number of clusters instead of making it a user-defined hyperparameter and unlike other clustering algorithms, AP is not dependent on the initialization step [[14]](https://paperpile.com/c/NPPxEM/OxXw). AP works by recursively passing real-valued messages between each data point till they converge, and based on these converged values, the algorithm establishes the clusters and assigns each cluster an "exemplar data point" which serves as an ideal representative of that cluster [[15]](https://paperpile.com/c/NPPxEM/ZrGQ) . Furthermore, AP clustering methods have shown success in clustering textual data[[16,17]](https://paperpile.com/c/NPPxEM/vxqy+FX6U). We use distance matrices computed by calculating the pairwise edit distances in the previous section as a divergence metric for AP.

Once the clusters are computed, we determine the median cluster size and identify clusters larger than the median cluster size. These clusters are designated as large clusters. We perform AP clustering within these clusters until their sizes drop below the previously determined median cluster size of the AP algorithm converges, and no more clustering can be performed.

We then check for outliers within each cluster using isolation forest and local outlier factors (LOF). If a data point within a cluster is determined to be an outlier using either of the methods, it is removed from that cluster and labeled as a new cluster with just that data point. We implemented isolation forest using the 'isolation.forest' package in the R programming language[[18]](https://paperpile.com/c/NPPxEM/5rKu). The number of trees (trees) using the commonly used heuristic is determined by computing the square root of the number of columns in the data matrix and setting the dims argument to 3, as suggested for numeric datasets in the package documentation. The isolation scores are calculated for each data point within a cluster, and if the isolation score is above 0.5, that data point is deemed an outlier. [[19]](https://paperpile.com/c/NPPxEM/ibnD)

Similarly, for LOF, we use the lof function within the "dbscan" package in the R programming language . To calculate the LOF value of this function, we needed to specify the number of nearest neighbors used to define the local neighborhood of a data point ("minPts"). We compute the lof values for "minds" ranging from 2 (clusters need to have more than one element to have an outlier) to the size of cluster minus 1. We compute the median LOF for each data point in the cluster, and if this value is above 1, that point is deemed an outlier.

Upon completing the outlier analysis, we implement the pipeline's second stage, which standardizes each cluster member. To achieve this, we go through each cluster once again and identify the closest term from the list of standardized tumors (WHO 5th edition, WHO all editions, and NCIT) using the edit distance we have used thus far in the pipeline. If a majority of the tumor names in a cluster are close to a specific standardized term, then all the tumor names are mapped to that standardized term. In case there is a draw, where two or more standardized terms are equally represented in a cluster, then each tumor name within that cluster is assigned to its closest (based on the edit distance used so far) matching standardized tumor name. The text-based matching pipeline is described in figure 2. The following section will discuss the pipeline that standardizes the clinical trial tumors based on text-embeddings.

**Fig 2: Text Match Pipeline using Edit Distances  
**

***Text Embedding Analysis: Closest match in Embedding Space***

The methods in the previous section employed distances to compare texts, these methods primarily focus on syntactical differences to quantify the differences between texts. In this section , the standardization methods are instead based on text embeddings (or word embeddings), which can also be used for comparing texts. Text embeddings are low dimensional numeric vector space representations of unstructured text data. Unlike edit distances, text embeddings focus on capturing the semantic and contextual meaning of the input text they encode;consequently in the embedding vector space, texts with similar meanings should have embeddings close to each other and texts which differ in their meaning should be further apart [[20]](https://paperpile.com/c/NPPxEM/SePY) [[21]](https://paperpile.com/c/NPPxEM/AIyW) [[22]](https://paperpile.com/c/NPPxEM/WbHn) [[23]](https://paperpile.com/c/NPPxEM/bKvK). Text-embeddings have been used in various applications such as developing search engines[[24,25]](https://paperpile.com/c/NPPxEM/fu7E+8uDZ) , text clustering [[26]](https://paperpile.com/c/NPPxEM/NP5Q)and classification[[27]](https://paperpile.com/c/NPPxEM/73kP), recommender systems[[28]](https://paperpile.com/c/NPPxEM/1mjP), and anomaly detection[[29]](https://paperpile.com/c/NPPxEM/ZV6P). Text-embeddings can be generated by natural language processing models such as Word2Vec, GloVE, FastText or through large language models such as BERT, GPT, ELMO [[23]](https://paperpile.com/c/NPPxEM/bKvK). In this paper, we generate text-embeddings from the following two embedding models offered by Open AI: text-embedding-ada-002 (referred as ADA002 in rest of the text) and text-embedding-3-large (referred as LTE-3 in rest of the text) [[30]](https://paperpile.com/c/NPPxEM/l3Uz). LTE-3 is a more recent and improved model and generates embeddings that have 3072 dimensions , whereas ADA002 generates embeddings consisting of 1536 dimensions.

We use both LTE-3 and ADA002 to generate embeddings for all the tumors identified in clinical trials, each term in every edition of the WHO database, and all the terms in the NCIT database. Once these sets of embeddings were generated, a simple method to standardize the tumor names in clinical trials would be to calculate their Euclidean distances from each standardized tumor term in the WHO and NCIT database and identify the nearest matching term. We did this using each set of embeddings and standardized the clinical trials tumor with respect to the 5th edition of WHO Tumors, all editions (3rd-5th) WHO Tumors, and NCIT Tumors. These results are reported in file XXXXXXX.

***Text Embedding Analysis: Open AI Embeddings and Clustering*** Similar to how we used edit distances as a divergence metric for AP clustering and then performed standardization on each cluster, the Euclidean distance between terms in the embedding space can also be used as a divergence metric in AP clustering. However, calculating the pairwise Euclidean distance in the embedding space for all the tumor terms in clinical trials, WHO, and NCIT database is computationally expensive as the embedding space has a large number of dimensions for both ADA002 (1536 dimensions) and LTE-3 (3072 dimensions) embeddings. Thus, we perform principal component analysis (PCA) on each set of embeddings, and it should be noted that before computing the PCA, we also have to consider the terms if we are using only the 5th edition of WHO tumors or all the editions, thus we generate four sets PCA transformed embeddings, and their dimensions are listed in Table 4. For each case, we only retain the minimum number of principal components that explain 80% of the variance in the data.

**Table 4: PCA dimensions for each embeddings based on WHO editions**

| Tumor Terms | Dimensions for PCA +ADA002 | Dimensions for PCA + LTE-3 |
| --- | --- | --- |
| CT + NCIT + WHO 5th Edition | 136 | 178 |
| CT + NCIT + WHO All Edition | 141 | 185 |

With the four sets of PCA-transformed embeddings, we calculate the pairwise Euclidean distance that can be used for AP clustering. Once AP clustering is completed, we identify the large clusters by computing the z-scores for each cluster based on the number of cluster members. We set the z-score threshold at 2.5, i.e., any clusters with a z-score greater than 2.5 were designated as large clusters. Using the threshold z-score of 2.5, we are able to determine the maximum number of members per cluster, and then for each cluster we designated as large, we perform AP within these clusters till either their sizes are below or equal to the maximum number of cluster members or the AP clustering algorithm converges. In this embedding-based AP clustering, we did not use median cluster size as the threshold for determining the large clusters as we did in edit-distance-based AP clustering because we noticed that the median cluster size in embedding-based AP clustering determined several clusters as large, which should not be considered large as those clusters elements belonged together. The z-score-based method flagged fewer such clusters as large and thus was a better way to identify large clusters in embedding-based AP clustering.

After the clustering is completed, we compute isolation forest and LOF analysis to determine outliers within each cluster just as we did in the case for edit-distance based AP clustering. The only change we made in this case was in the isolation forest analysis we fixed the number of trees argument to 100 as it is the value recommended for convergence by Liu et al. 2008 [(Liu et al. 2008)](https://paperpile.com/c/NPPxEM/vpqB) in their original introduction of the isolation forest algorithm. The remaining hyperparameters for both isolation forest and LOF analysis were kept the same as they were in the case of edit-distance based AP clustering. Following the outlier detection step, we iterate through each cluster and determine the standardized term (WHO 5th edition, WHO all edition, or NCIT) closest to each cluster element by computing the euclidean distance in the embedding space (not PCA transformed space). If a majority of the tumor names in a cluster are close to a specific standardized term, then all the tumor names are mapped to that standardized term. In case there is a draw, where two or more standardized terms are equally represented in a cluster, then each tumor name within that cluster is assigned to its closest matching standardized tumor name. The final standardization process is similar to the standardization process discussed in the edit-distance based AP clustering, however, instead of edit distances , we used the euclidean distance in the embedding space to determine the closest standardized terms. In table 5 we compare and contrast the number of clusters we got from AP clustering when we used the text-embeddings and edit-distances.

**Table 5: Number of clusters from AP clustering under embedding and text-matching based methods**.

| Basis | Affinity Propagation Clustering Divergence Metric | Number of Clusters for CT Terms, NCIT Terms, All Editions WHO Terms | Number of Clusters for CT Terms, NCIT Terms, 5th Edition WHO |
| --- | --- | --- | --- |
| Text Match | Cosine | 1040 | 967 |
| Text Match | Levenshtein | 2020 | 1808 |
| Text Match | Jarro Winkler | 1965 | 1785 |
| Embedding | ADA002 + Euclidean Dist | 3790 | 3456 |
| Embedding | LTE-3 + Euclidean Dist | 3894 | 3427 |

In addition to AP clustering, we also implemented KMeans clustering on the PCA transformed embeddings to standardize the tumor names in clinical trials. Unlike AP clustering, the KMeans algorithm requires the user to define “K” or the numbers of clusters to be formed as a hyperparameter [(Wu 2012)](https://paperpile.com/c/NPPxEM/wp0i). Since we neither have any a priori information on the types of tumors that are present in the clinical trials database or have any means to classify the tumors into based on their tissue or molecular subtypes, we needed to use other computational methods to decide on a value for the number of clusters to provide as an input to the KMeans algorithm. Thus to determine the number of clusters, we computed a commonly used cluster performance metric known as silhouette coefficient [(Shahapure and Nicholas 2020)](https://paperpile.com/c/NPPxEM/rUdk). After the clustering is completed, silhouette coefficient is computed for each data point and it ranges from [-1, 1]. A silhouette coefficient of 1 signifies that the data point is well matched to other elements in its own cluster and poorly matched to members of neighboring clusters [(Shutaywi and Kachouie 2021)](https://paperpile.com/c/NPPxEM/HATf). A score of 0 indicates that the data point is at the decision boundary of neighboring clusters and a score of -1 indicates that the data point is in the wrong cluster [(Shutaywi and Kachouie 2021)](https://paperpile.com/c/NPPxEM/HATf). Thus, higher the silhouette coefficient the better is the clustering. Intuitively, a higher silhouette coefficient for a data point represents high cohesion of that data point with rest of the cluster members and high separation from members of neighboring clusters.

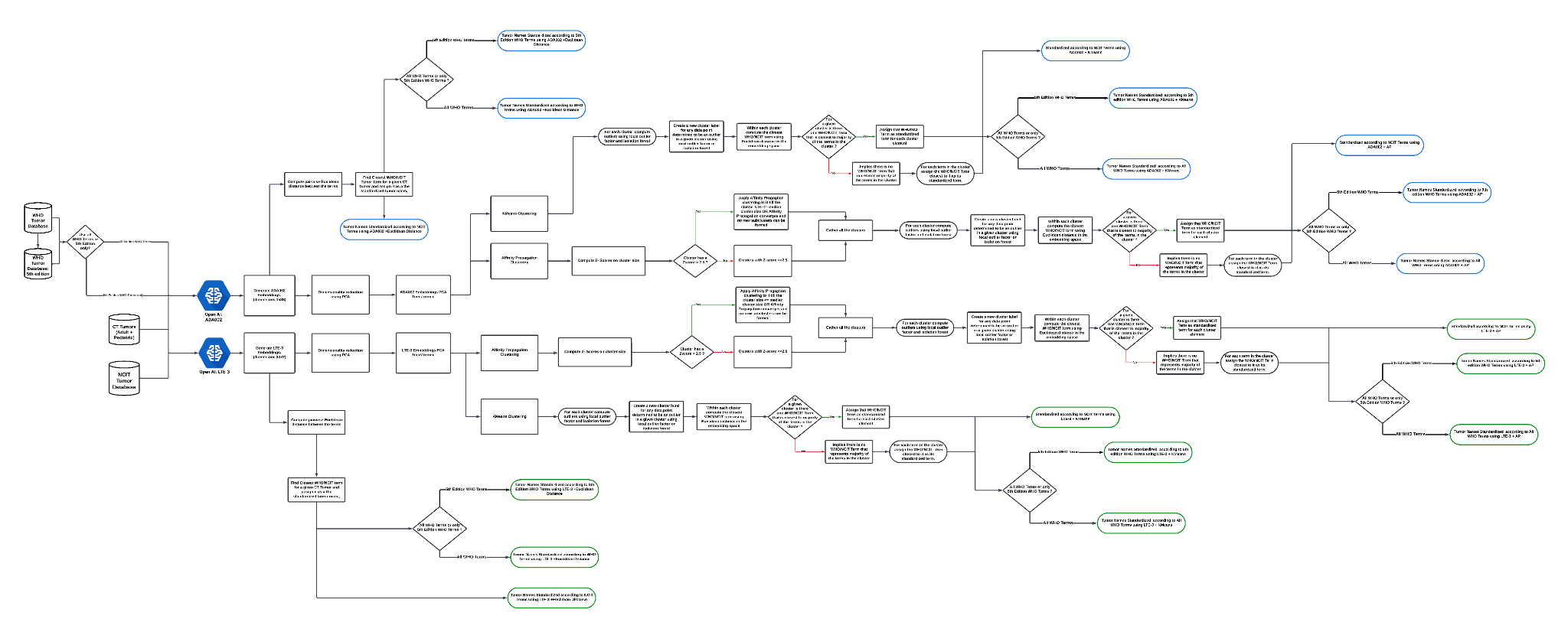
For several values of “K”, we perform KMeans clustering and calculate the silhouette coefficient for each data point. For each cluster we can calculate the average silhouette coefficient which indicates the performance for that cluster, and further taking the average of all the averaged silhouette coefficients provides a metric that can be used to evaluate the overall clustering performance of the KMeans algorithm for a given value of “K”, with higher value being a better clustering performance. We refer to this metric as the mean silhouette score. Using this method, we evaluated the mean silhouette score for various values of “K” under both ADA002 and LTE-3 embeddings and when only the 5th edition of WHO tumors or all editions of WHO tumors were considered in the pipeline along with the NCIT terms. In figure 3, we plot the mean silhouette score for each of these cases.

**Figure 3: Average Silhouette Score vs Number of clusters (K) used in KMeans. a. ADA002 when all editions of WHO terms are used. b. ADA002 when 5th edition of WHO terms are used c. LTE-3 when all editions of WHO terms are used d. LTE-3 when 5th editions of WHO terms are used**

****

Once the clustering is completed using KMeans, we follow the exact same steps for outlier detection (isolation forest and LOF analysis) and standardization (compute euclidean distances with standardized term and identify the closest match) as we did for AP clusters and maintain the exact hyperparameter configurations. The entire text-embedding based standardization pipeline is displayed in figure 5.

**Figure 5: Text Match Pipeline using ADA002 and LTE-3 Embeddings**

****

***2.3.3.2 Clustering with KMeans and Standardization to WHO Terms for ADA2.0 and V-3 Large Embeddings***

***2.3.3.3 Clustering with Affinity Propagation Clustering and Standardization to WHO Terms for ADA2.0 and V-3 Large Embeddings***

**Results:**

**Discussion:**

**Conclusion:**

**Figures**

**Fig4: Silhouette Score vs cluster size (for K-means) using ADA002 and LTE-3 Embeddings**

**Appendix**

**A1: Demonstration of comparing strings using edit distances**In order to illustrate the concept of how strings can be compared using edit distances, consider the following two strings:

String 1: Breast Cancer

String 2: Brain Cancer

Let us now suppose, we can make the following operations: deletions, substitution, and insertion to transform string 1 to string 2. Then we can proceed to transform String 1 to String 2 in the following ways:  
  
Table A1: step by step demonstration of comparing strings using edit distances.

| Method 1 | Method 1 operation | Method 2 | Method 2 operation |
| --- | --- | --- | --- |
| Breast Cancer |  | Breast Cancer |  |
| Brast Cancer | Delete ‘e’ | Braast Cancer | Substitute ‘e’ with ‘a’ |
| Bras Cancer | Delete ‘t’ | Braist Cancer | Substitute ‘a’ with ‘i’ |
| Bra Cancer | Delete ‘s’ | Braint Cancer | Substitute ‘s’ with ‘n’ |
| Brai Cancer | Insert ‘i’ | Brain Cancer | Delete ‘t’ |
| Brain Cancer | Insert ‘n’ |  |  |

We can observe from Table A1 that the two methods transform string 1 to string 2, but method 2 employs fewer steps to achieve this objective. Furthermore, with the given set of operations the minimum number of steps required to transform string 1 to string 2 is four. This is achieved by method 2 , thus the edit distance between string 1 and string 2 is four. The set of operations that were used to transform the strings and method with which we calculated the edit distance is also known as the Levenshtein distance.

**References**

1 [Bray F, Laversanne M, Sung H, *et al.* Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J Clin*. 2024;74:229–63.](http://paperpile.com/b/NPPxEM/Weij)

2 [Siegel RL, Giaquinto AN, Jemal A. Cancer statistics, 2024. *CA Cancer J Clin*. 2024;74:12–49.](http://paperpile.com/b/NPPxEM/6kRx)

3 [Matt GY, Sioson E, Shelton K, *et al.* St. Jude Survivorship Portal: Sharing and Analyzing Large Clinical and Genomic Datasets from Pediatric Cancer Survivors. *Cancer Discov*. 2024;14:1403–17.](http://paperpile.com/b/NPPxEM/tUKD)

4 [Aristizabal P, Winestone LE, Umaretiya P, *et al.* Disparities in Pediatric Oncology: The 21st Century Opportunity to Improve Outcomes for Children and Adolescents With Cancer. *Am Soc Clin Oncol Educ Book*. 2021;41:e315–26.](http://paperpile.com/b/NPPxEM/FRbF)

5 [Hunger Stephen P., Mullighan Charles G. Acute Lymphoblastic Leukemia in Children. *N Engl J Med*. ;373:1541–52.](http://paperpile.com/b/NPPxEM/68hp)

6 [Laetsch TW, DuBois SG, Bender JG, *et al.* Opportunities and Challenges in Drug Development for Pediatric Cancers. *Cancer Discov*. 2021;11:545–59.](http://paperpile.com/b/NPPxEM/NRZI)

7 [Renfro LA, Ji L, Piao J, *et al.* Trial Design Challenges and Approaches for Precision Oncology in Rare Tumors: Experiences of the Children’s Oncology Group. *JCO Precis Oncol*. 2019;3. doi:](http://paperpile.com/b/NPPxEM/Lq9U) [10.1200/PO.19.00060](http://dx.doi.org/10.1200/PO.19.00060)

8 [Rivers Z, Hyde B, Ronski K, *et al.* Exploring Barriers to Pediatric Cancer Clinical Trials: The Role of a Networked, Just-in-Time Study Program. *Clin Ther*. 2023;45:1148–50.](http://paperpile.com/b/NPPxEM/5cZt)

9 [National Institutes of Health Clinical Trials Registry. ClinicalTrials.gov.](http://paperpile.com/b/NPPxEM/hYaH) <https://clinicaltrials.gov/> [(accessed 26 August 2024)](http://paperpile.com/b/NPPxEM/hYaH)

10 [Zarin DA, Tse T, Williams RJ, *et al.* The ClinicalTrials.gov Results Database — Update and Key Issues. *N Engl J Med*. 2011;364:852–60.](http://paperpile.com/b/NPPxEM/Hrsy)

11 [Siegel RL, Miller KD, Wagle NS, *et al.* Cancer statistics, 2023. *CA Cancer J Clin*. 2023;73:17–48.](http://paperpile.com/b/NPPxEM/GzPE)

12 [Snášel V, Keprt A, Abraham A, *et al.* Approximate String Matching by Fuzzy Automata. *Man-Machine Interactions*. Springer Berlin Heidelberg 2009:281–90.](http://paperpile.com/b/NPPxEM/rIfO)

13 [van der Loo M. Stringdist: Approximate string matching, fuzzy text search, and string distance functions. CRAN: Contributed Packages. 2013.](http://paperpile.com/b/NPPxEM/MKyW)

14 [Frey BJ, Dueck D. Clustering by passing messages between data points. *Science*. 2007;315:972–6.](http://paperpile.com/b/NPPxEM/OxXw)

15 [Kitahara YFGI. Fast Algorithm for Affinity Propagation. In: Walsh T, ed. *Twenty-Second International Joint Conference on Artificial Intelligence*. IJCAI/AAAI :2238–43.](http://paperpile.com/b/NPPxEM/ZrGQ)

16 [Shi XH, Guan RC, Wang LP, *et al.* An incremental affinity propagation algorithm and its applications for text clustering. *2009 International Joint Conference on Neural Networks*. IEEE 2009.](http://paperpile.com/b/NPPxEM/vxqy)

17 [Shailendra Kumar Shrivastava, J.L.Rana, R.C.Jain. Text document clustering based on phrase similarity using affinity propagation. *International Journal of Computer Applications*. 2013;61. doi:](http://paperpile.com/b/NPPxEM/FX6U) [10.5120/10032-5077](http://dx.doi.org/10.5120/10032-5077)

18 [Cortes D. isotree: Isolation-Based Outlier Detection. CRAN: Contributed Packages. 2019.](http://paperpile.com/b/NPPxEM/5rKu)

19 [Hahsler M, Piekenbrock M. Dbscan: Density-based spatial clustering of applications with noise (DBSCAN) and related algorithms. CRAN: Contributed Packages. 2015.](http://paperpile.com/b/NPPxEM/ibnD)

20 [Morris J, Kuleshov V, Shmatikov V, *et al.* Text embeddings reveal (almost) as much as text. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics 2023.](http://paperpile.com/b/NPPxEM/SePY)

21 [Mikolov T. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:13013781*. Published Online First: 2013.](http://paperpile.com/b/NPPxEM/AIyW)

22 [Incitti F, Urli F, Snidaro L. Beyond word embeddings: A survey. *Inf Fusion*. 2023;89:418–36.](http://paperpile.com/b/NPPxEM/WbHn)

23 [Khattak FK, Jeblee S, Pou-Prom C, *et al.* A survey of word embeddings for clinical text. *J Biomed Inform*. 2019;100S:100057.](http://paperpile.com/b/NPPxEM/bKvK)

24 [Gökçe O, Prada J, Nikolov NI, *et al.* Embedding-based scientific literature discovery in a text editor application. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics 2020.](http://paperpile.com/b/NPPxEM/fu7E)

25 [Mai G, Janowicz K, Yan B. Combining text embedding and knowledge graph embedding techniques for academic search engines. In: Key-Sun Choi, Luis Espinosa Anke, Thierry Declerck, Dagmar Gromann, Jin-Dong Kim, Axel-Cyrille Ngonga Ngomo, Muhammad Saleem, Ricardo Usbeck, ed. *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018)*. CEUR 2018:77–88.](http://paperpile.com/b/NPPxEM/8uDZ)

26 [Mehta V, Bawa S, Singh J. WEClustering: word embeddings based text clustering technique for large datasets. *Complex Intell Systems*. 2021;7:3211–24.](http://paperpile.com/b/NPPxEM/NP5Q)

27 [Stein RA, Jaques PA, Valiati JF. An analysis of hierarchical text classification using word embeddings. *Inf Sci* . 2019;471:216–32.](http://paperpile.com/b/NPPxEM/73kP)

28 [Musto C, Semeraro G, de Gemmis M, *et al.* Learning word embeddings from Wikipedia for content-based recommender systems. *Lecture Notes in Computer Science*. Cham: Springer International Publishing 2016:729–34.](http://paperpile.com/b/NPPxEM/1mjP)

29 [Pande A, Ahuja V. WEAC: Word embeddings for anomaly classification from event logs. *2017 IEEE International Conference on Big Data (Big Data)*. IEEE 2017:1095–100.](http://paperpile.com/b/NPPxEM/ZV6P)

30 [New embedding models and API updates.](http://paperpile.com/b/NPPxEM/l3Uz) <https://openai.com/index/new-embedding-models-and-api-updates/> [(accessed 4 September 2024)](http://paperpile.com/b/NPPxEM/l3Uz)

**ntress=100** [**https://www.rdocumentation.org/packages/isotree/versions/0.6.1-1/topics/isolation.forest**](https://www.rdocumentation.org/packages/isotree/versions/0.6.1-1/topics/isolation.forest)