A post-processing algorithm for building longitudinal medication dose data from extracted medication information using natural language processing from electronic health records

Elizabeth McNeer, MS[1], Cole Beck, BS[1], Hannah L. Weeks, BS[1], Michael L. Williams, BS[1], Cosmin Adrian Bejan, PhD[2], Joshua C. Denny, MD, MS[2,3], Leena Choi, PhD[1]

[1]Department of Biostatistics, Vanderbilt University Medical Center, Nashville, TN, USA
[2]Department of Biomedical Informatics, Vanderbilt University Medical Center, Nashville, TN, USA
[3]Department of Medicine, Vanderbilt University Medical Center, Nashville, TN, USA

Keywords: medication extraction; electronic health records; natural language processing; post processing algorithm.

ABSTRACT

Objective

We developed a post-processing algorithm to convert raw natural language processing output from electronic health records into a usable format for analysis. This algorithm was specifically developed for creating datasets that can be used for medication-based studies.

Materials and Methods

The algorithm was developed using output from two natural language processing systems, MedXN and medExtractR. We extracted medication information from deidentified clinical notes from Vanderbilt's electronic health record system for two medications, tacrolimus and lamotrigine, which have widely different prescribing patterns. The algorithm consists of two parts. Part I parses the raw output and connects entities together and Part II removes redundancies and calculates dose intake and daily dose. We evaluated both parts of the algorithm by comparing to gold standards that were generated using approximately 300 records from 10 subjects for both medications and both NLP systems.

Results

Both parts of the algorithm performed well. For MedXN, the F-measures for Part I were at or above 0.94 and for Part II they were at or above 0.98. For medExtractR the F-measures for Part I were at or above 0.98 and for Part II they were at or above 0.91.

Discussion

Our post-processing algorithm is useful for drug-based studies because it converts NLP output to analyzable data. It performed well, although it cannot handle highly

complicated cases, which usually occurred when a NLP incorrectly extracted dose information. Future work will focus on identifying the most likely correct dose when conflicting doses are extracted on the same day.

Introduction

Natural language processing (NLP) systems extract information from unstructured text and convert it into a structured format. The development of NLP systems for extracting information from electronic health records (EHRs) has led to great opportunities for performing diverse research using EHRs by providing critical pieces of data. Many NLP systems have been developed specifically for clinical research. For example, cTAKES[1], MetaMap[2], and MedLEE[3] are NLP systems for general purpose extraction of clinical information. In 1996, Evans et al. used the CLARIT system to extract medication names and dosage information from clinical narrative text.[4] Their system extracted medication dosage information with about 80% accuracy. More recently, MedEx[5], CLAMP[6], MedXN[7], and medExtractR[8], have been developed to extract medication information from EHRs more accurately, which is useful for drug-based studies.

The raw output from some of these systems is not directly usable and requires a post-processing step to convert the extracted information into an appropriate data form for further analysis depending on the research goal. For an example of medication extraction, the entities (or attributes) of the raw output from NLP systems should be associated with corresponding medication names to make dose data, such as dose given intake or daily dose, that can be used for further analysis. Some NLP systems have a built-in post-processing step as a part of the system. For example, the final step in the system built by Patrick et al. for the 2009 i2b2 medication extraction challenge was equipped with a medication entry generator for assembling medication events based on the relationships between components established in previous steps.[9] Also, the Lancet

system developed by Li et al. included a supervised machine learning classifier that attempted to associate a medication name with the correct entities.[10] MedEx[5] and Clamp[6] also include a post-processing step that connects entities with each drug name in a data structure.

However, to the best of our knowledge, the validation of post-processing algorithms for these NLP systems has not been reported separately from the overall evaluation of the NLP systems in the literature, although they may be unofficially validated during the development phase. Correctly connecting entities with both drug name and with each other to obtain medication dose (e.g., dose given intake or daily dose) can be challenging and error prone if the prescription pattern is complex. Developing and validating a post-processing algorithm separately from the main NLP system would allow for easier identification of error sources and more efficient improvement of relevant parts of the system.

Thus, we developed algorithms that process the raw output from two medication extraction systems, MedXN and medExtractR, which performed best out of 4 NLP systems previously tested.[8] The post-processing algorithm we developed is divided into two parts. Part I processes the raw output from the NLP system and connects entities together. Part II removes redundant data entries anchored at each note or date identifier and calculates dose given intake and daily dose. These measurements are crucial for medication-based studies such as population pharmacokinetic and pharmacodynamic or pharmacogenomic studies.

Methods

Data source

To develop a post-processing algorithm we used two medications, tacrolimus and lamotrigine, whose prescription patterns vary from simple to complicated. For each medication, we defined a patient cohort separately using patient records in a de-identified database of clinical records derived from Vanderbilt's EHR system. For tacrolimus data, we used the same cohort (n=466) used in previous studies[11], who were treated with tacrolimus after renal transplant. For lamotrigine data, we first identified patient records with 'lamotrigine' and 'Lamictal' (the brand name of lamotrigine) and an ICD-9-CM or ICD-10-CM (The International Classification of Diseases, Ninth and Tenth Revision, Clinical Modification) billing code for epilepsy before October 3, 2017. We further refined the cohort by selecting patients who had their first lamotrigine level between 18 and 70 years of age and at least 3 drug levels and 3 doses, which yielded the final cohort of 305 subjects. For each subject of each cohort, we identified all clinical notes generated on the same dates when drug concentration laboratory values were available, from which medication dosing information was extracted using both MedXN and medExtractR.

Medication entities extraction using existing NLP systems

MedXN

The Medication Extraction and Normalization (MedXN) system was designed to extract medication information from clinical notes and convert it into an RxNorm concept unique identifier (RxCUI). [7] This system identifies medication names and attributes,

such as dosage, strength, and frequency, in clinical notes using the RxNorm dictionary and regular expressions. The attributes associated with a medication name are combined together in the RxNorm standard order and normalized to a specific RxCUI.

medExtractR

MedExtractR is a medication extraction algorithm built using the R programming language, and is more targeted than other more general purpose NLP systems.[8] Given a list of drug names to search for, medExtractR creates a search window around each identified drug mention within a clinical note in which to search for related drug entities. The system shortens the search window when a medication name which is not of interest appears, to avoid extracting incorrect or irrelevant information. By default, the list of unrelated drug names is based on the RxNorm library supplemented with common abbreviations. Some drug entities are identified and extracted based on matching expressions in manually curated entity-specific dictionaries, including frequency, intake time, and dose change (keywords to indicate that a regimen may not be current). For the remaining entities, including strength, dose amount, dose (i.e., dose given intake), and time of last dose, regular expressions are used to identify common patterns for how these entities are written. Function arguments can be used to optimize drug entity extraction for a given set of clinical notes. Examples include specifying the maximum edit distance for approximate drug name matching or the length of the search window.

Description of post-processing algorithms

Converting the output from MedXN and medExtractR into a form that can be

used for analysis requires post-processing. A simple post-processing method we considered formed all possible combinations of entities. For example, using this simple method on a drug mention with two strengths, two dose amounts, and two frequencies would have resulted in eight combinations of strengths, dose amounts, and frequencies. However, we know most of these combinations are incorrect, and hence we developed a more sophisticated post-processing algorithm to give us more useful data for analysis.

Part I processes the raw NLP output and then pairs the parsed entities, as outlined along with some illustrative examples in Figure 1. The algorithm begins with a single file that is the raw output from a NLP system, from which a drug name and its entities are isolated and converted to a standardized form [**Step 1**]. The entities include strength, dose amount, route, frequency, duration, dose change, and dose given intake ("dose"). The dose change and dose entities are specific to medExtractR. Dose change includes key words such as "increase" that indicate that the dose isn't a current dose. Dose is an aggregate total dose given intake when dose amount information is not found. Note that Figure 1 only includes the strength, dose, route, and frequency entities for simplicity. The standardized form includes a row for each drug mention and columns for the entities anchored to that mention. Next, any records with invalid drug names (i.e., names of drugs which are not part of the study) are removed [**Step 2**]. To begin the pairing part of the algorithm, clusters (groups of entities which are close together) are formed in rows with competing entities (e.g., two frequencies). The clusters are formed when the start distance between entities exceeds a gap of some length. The recommended gap size of 32 was determined from checking a range of gap sizes in our preliminary data and choosing the one that resulted in the most reasonable pairing of entities. If an entity occurs once in

only one cluster, this entity is added to every cluster in that row [**Step 3**]. For example, Case 2 in Figure 1 shows that the strength of 100 mg in Cluster 1 is added to Cluster 2. If multiple entities occur within a single cluster, all combinations of entities are formed for that cluster [**Step 4**]. For example, Case 3 in Figure 1 shows that all combinations of the two strengths and three frequencies in Cluster 1 are formed. Based on entity order and position, each combination is examined and determined to be good or bad. Bad combinations are removed, while unassigned entities may be used to create a combination with missing data [**Step 5**]. For example in Case 3 in Figure 1, each of the strengths and frequencies in Cluster 1 will be assigned a "group": strength 1 (50 mg) and frequency 1 (qAM) are assigned group 1, strength 2 (100 mg) and frequency 2 (bid) are assigned group 2, and frequency 3 (tid) is assigned group 3. Because strength 1, frequency 1 pair and strength 2, frequency 2 pair within groups 1 and 2, rows 1 and 5 in this example are kept. Because frequency 3 has not been matched to other entities, row 7 is added with non-matched entities set to missing.

Part II removes redundant data entries anchored at note or date level for a given patient and calculates dose given intake and daily dose. An overview of steps along with some examples are presented in Figure 2. Note that Figure 2 only includes the strength, dose amount, and frequency entities for simplicity. Part II starts with the output from Part I. First, all character entities (e.g., strength, dose amount, frequency, and dose) are converted to numeric values [**Step 1**]. As part of that process, frequency values are standardized as strings. For example, the frequencies "twice a day", "twice daily", "bid", etc. are all standardized to the string "bid". Frequencies that give a time of day (e.g., "qam", "at noon", "in the evening", etc.), are assigned a numeric value of 1, and a new

entity “intaketime” is added to record this information, using the standardized strings “am”, “noon”, and “pm”. Rows that include only drug name are removed [**Step 2**]. If there are drug name changes in adjacent rows within the same note (e.g., lamotrigine to Lamictal), these rows are collapsed into one row if possible [**Step 3**]. This usually happens when a phrase such as “lamotrigine 100mg (also known as Lamictal) 3 tablets bid” is present in the original note, resulting in two rows of output for the same drug mention. In this example, one row has a drug name of lamotrigine, a strength of 100 and no dose amount or frequency while the next row has a drug name of Lamictal, a dose amount of 3 and a frequency of 2 with no strength. These two rows are combined, yielding a single row with a strength of 100, a dose amount of 3, and a frequency of 2. If a strength is present but dose amount is missing, dose amount is set to 1 [**Step 4**]. Next, information about strength, frequency, route, and duration is borrowed within the same note when possible [**Step 5**]. If strength, frequency, or route is missing and there is a unique strength, frequency, or route within the same note, that strength, frequency, or route is borrowed. If there is not a unique strength within the same note, the closest preceding strength is borrowed. If there is no preceding strength, the closest strength after the missing strength is borrowed. If there is not a unique frequency or route within the same note, the most common frequency or route within the note is borrowed. Duration is only borrowed within a dose sequence. For example, a row with an intake time of “am” and no duration could borrow the duration in the row after it if that row has an intake time of “pm”. Any records that are still missing strength are removed, since dose cannot be calculated in these cases [**Step 6**]. If records are still missing frequency or route, the most common frequency or route across all observations for that drug is imputed [**Step**

7]. Next, the dose given intake is computed by multiplying strength by dose amount. If there is not an intake time, the daily dose is computed by multiplying dose given intake by frequency. If there is an intake time (e.g., am, noon, or pm), the daily dose is calculated by adding the dose given intakes at each of the intake times [**Step 8**]. Finally, any redundancies are removed at date level and note level separately, yielding two datasets [**Step 9**].

Generation of training and test sets

We generated training and test sets for each medication (i.e., tacrolimus and lamotrigine) from medication entities extracted by each NLP system (i.e., MedXN and medExtractR). Each dataset included approximately 300 observations from 10 patients. Patients with complex data were over sampled. Complexity was determined by the presence of multiple clusters or clusters containing entities with conflicting values. These cases are often difficult to process and are likely to produce discrepant daily dose. For example, when two different strengths of 100 mg and 200 mg are associated with a lamotrigine mention, they are more difficult to process compared to only a unique strength of 100 mg associated with that drug mention since each strength must be paired with the correct dose amount and frequency. Thus, for each of the training and test sets, we randomly selected a greater number of complicated cases; six of the tacrolimus patients were chosen from patients with known complications as well as eight of the lamotrigine patients.

Making gold standard datasets

For each of the training and test sets, we generated three sets of gold standard datasets to test each part of the post-processing algorithm. Each gold standard dataset was manually generated to make the intended output for each part of the algorithm. That means that we generated "Gold Standard I" as the output of Part I if the five steps of the algorithm were correctly performed. Then, we generated "Gold Standard II–Date" and "Gold Standard II–Note" as the output data of Part II applied to the Gold Standard I at date level and at note level, respectively, if the nine steps of the algorithm were correctly performed. These were generated for each of the training and test sets and each of the medications (i.e., tacrolimus and lamotrigine), yielding a total of 12 sets of gold standard datasets for each NLP system.

Evaluation of algorithms

The algorithms were evaluated using recall, precision, and F1-measure. Recall is the proportion of the gold standard entities that were correctly identified by the algorithm. Precision is the proportion of the extracted entities that were correctly found in the gold standard. The F1-measure is defined as 2*(precision*recall)/(precision+recall).

Results

Performance

We evaluated each part of the algorithm separately. Table 1 presents the results for Part I. The algorithm performed well on both medications for both MedXN and medExtractR. The recall was slightly higher for tacrolimus than lamotrigine when using the MedXN

output (0.96 vs. 0.94), but the precision was slightly lower for tacrolimus (0.93 vs. 0.94). The F1-measures were the same for both medications (0.94). The algorithm performed better on the medExtractR output than the MedXN output. The tacrolimus recall/precision/F1-measure was 1.00/1.00/1.00 while lamotrigine was 0.98/0.98/0.98.

The results for Part II can be found in Table 2. Here we present the recall, precision, and F1-measure for both medications at the note and date level, and we consider the dose intake and daily dose. The note level collapsing performed very well with all F1-measures equal to 1 for dose intake for both medications and both NLP systems, and F1-measures ranging from 0.96 to 1 for daily dose. Date level collapsing also performed well with F1-measures ranging from 0.95 to 1 for dose intake and 0.91 to 1 for daily dose.

Table 1: Recall, precision, and F1-measures for Part I

	MedXN			medExtractR		
	Recall	Precision	F1	Recall	Precision	F1
Tacrolimus	0.96	0.93	0.94	1.00	1.00	1.00
Lamotrigine	0.94	0.94	0.94	0.98	0.98	0.98

Table 2: Recall, precision, and F1-measures for Part II (note/date)

		MedXN			MedExtractR		
		Recall	Precision	F1	Recall	Precision	F1
Dose Intake	Tacrolimus	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/0.92	1.00/0.96
	Lamotrigine	1.00/1.00	1.00/1.00	1.00/1.00	1.00/0.98	1.00/0.93	1.00/0.95
Daily Dose	Tacrolimus	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/0.92	1.00/0.96
	Lamotrigine	0.99/0.98	0.99/0.98	0.99/0.98	0.96/0.94	0.97/0.89	0.96/0.91

Error analyses and examples of challenges

When making the Gold Standard I, we sometimes paired entities based on their positions in the original note. This only happened for complicated cases when the number of each entity was not equal (e.g., three strengths and two dose amounts), which caused difficulty on our judgement for pairing; hence, we looked at the position in the original

note to make the pairing decision. Since Part I only used position to make the clusters but did not use information about the position of the entities when pairing, rather it used order, this sometimes resulted in disagreement between the gold standard and the algorithm. Table 3 presents an example of this case. In this example, there are three strengths and four frequencies in the MedXN output. The algorithm pairs them in order, but the Gold Standard I paired them based on position.

Another challenge occurred when the NLP extracted incorrect information. This could have occurred because of a misspelling, missing spaces, or an uncommon abbreviation of a drug name, which caused the NLP to extract entities anchored to the wrong drug mention. This results in several extra strengths, dose amounts, or frequencies, making for a very complicated cluster structure. Table 4 illustrates this example. Here, all of the information that was extracted by MedXN was incorrect because of the missing spaces between the frequencies and the next drug name.

Also, the gap size of 32 occasionally caused issues for pairing in Part I. As an example, Table 5 shows that the first cluster includes the strength of 100 mg and the frequency of bid, the second cluster includes the first dose amount of 2 and the frequency of daily, and the third cluster includes the second dose amount of 2, so the output has three rows. However, because there were two dose amounts and two frequencies, these entities were paired in order in the Gold Standard I.

In Part II, challenges occurred when the morning dose came after the evening dose. The algorithm only treated dose sequences correctly when the morning dose came before the evening dose.

Discussion

Detailed medication dose information is often required to perform medication-based population studies. Medication dose data can be obtained from a structured data source or an unstructured data source such as clinical notes in EHRs. To extract medication dose information from unstructured text, a specialized algorithm such as a natural language processing system is commonly used. However, the output of NLP systems is often not in a form that is useful for analysis. We developed a post-processing algorithm to address this issue. Our algorithm consists of two parts to parse raw NLP output, connect medication names and attributes, and eliminate redundant information.

Our post-processing algorithm was developed using the output from two NLP systems, MedXN and medExtractR. Our algorithm performed reasonably well to process the output from both MedXN (F-measures Part I: ≥ 0.94; Part II: ≥ 0.98) and medExtractR (F-measures Part I: ≥ 0.98; Part II: ≥ 0.91). We tested the algorithm using two medications that have widely different prescribing patterns, but it should be tested using other medications. We also have Part I written for two other NLP systems, MedEx and CLAMP, but we have not yet tested the algorithm using a gold standard for these systems. We are in the process of incorporating a few changes to the algorithm, such as pairing entities based on distance in Part I in order to improve the performance.

The goal of the post-processing algorithm was to convert all the extracted information from an NLP system into a usable format. However, this may result in conflicting doses on the same day, which will need to be resolved before the data can be used for medication-based studies such as pharmacokinetic or pharmacodynamic studies.

Future work will focus on identifying the most likely correct dose when conflicting doses are present.

Acknowledgement

This work was supported by National Institutes of Health (NIH)/R01 GM124109.

Conflict of Interest

None declared.

References

1 Savova GK, Masanz JJ, Ogren PV, *et al.* Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc* 2010;**17**:507–13. doi:10.1136/jamia.2009.001560

2 Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc* 2010;**17**:229–36. doi:10.1136/jamia.2009.002733

3 Friedman C, Alderson PO, Austin JHM, *et al.* A General Natural-language Text Processor for Clinical Radiology. *J Am Med Inform Assoc* 1994;**1**:161–74. doi:10.1136/jamia.1994.95236146

4 Evans DA, Brownlow ND, Hersh WR, *et al.* Automating concept identification in the electronic medical record: an experiment in extracting dosage information. *Proc Conf Am Med Inform Assoc AMIA Fall Symp* 1996;:388–92.

5 Xu H, Stenner SP, Doan S, *et al.* MedEx: a medication information extraction system for clinical narratives. *J Am Med Inform Assoc* 2010;**17**:19–24. doi:10.1197/jamia.M3378

6 Soysal E, Wang J, Jiang M, *et al.* CLAMP – a toolkit for efficiently building customized clinical natural language processing pipelines. *J Am Med Inform Assoc* 2017;**25**:331–6. doi:10.1093/jamia/ocx132

7 Sohn S, Clark C, Halgrim SR, *et al.* MedXN: an open source medication extraction and normalization tool for clinical text. *J Am Med Inform Assoc* 2014;**21**:858–65. doi:10.1136/amiajnl-2013-002190

8 Weeks HL, Beck C, McNeer E, *et al.* medExtractR: A medication extraction algorithm for electronic health records using the R programming language. *bioRxiv* 2019.

9 Patrick J, Li M. High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge. *J Am Med Inform Assoc* 2010;**17**:524–7. doi:10.1136/jamia.2010.003939

10 Li Z, Liu F, Antieau L, *et al.* Lancet: a high precision medication event extraction system for clinical text. *J Am Med Inform Assoc* 2010;**17**:563–7. doi:10.1136/jamia.2010.004077

11 Birdwell KA, Grady B, Choi L, *et al.* The use of a DNA biobank linked to electronic medical records to characterize pharmacogenomic predictors of tacrolimus dose requirement in kidney transplant recipients. *Pharmacogenet Genomics* 2012;**22**:32–42. doi:10.1097/FPC.0b013e32834e1641

Table 3: Example of Gold Standard I paired by position disagreeing with the Part I output

MedXN Output:				
ID5_2013-11-07_note1.txt|Lamictal::1482::1490|196502|200 mg::1534::1540`100 mg::1557::1563`100 mg::1576::1582|1::1492::1493|tablet::1494::1500|mouth::1504::1509|twice a day::1510::1521`with breakfast::1541::1555`with lunch::1564::1574`with dinner::1583::1594				
Gold Standard I:				
ID5_2013-11-07_note1.txt	Lamictal		1	twice a day
ID5_2013-11-07_note1.txt	Lamictal	200 mg	1	with breakfast
ID5_2013-11-07_note1.txt	Lamictal	100 mg	1	with lunch
ID5_2013-11-07_note1.txt	Lamictal	100 mg	1	with dinner
Part I Output:				
ID5_2013-11-07_note1.txt	Lamictal	200 mg	1	twice a day
ID5_2013-11-07_note1.txt	Lamictal	100 mg	1	with breakfast
ID5_2013-11-07_note1.txt	Lamictal	100 mg	1	with lunch
ID5_2013-11-07_note1.txt	Lamictal		1	with dinner

Table 4: Example of NLP extracting incorrect information causing difficulty in pairing entities

Original Note:				
lamotrigine 200 mg tablet (Also Known As Lamictal) 1 tablet po bidSymbicort 80 mcg-4.5 mcg/actuation HFA aerosol inhaler 2 puffs from the inhaler twice a day prncetirizine 10 mg by mouth once daily prn allergiesibuprofen 800 mg by mouth every 6-8 hours as needed for painKeppra 800 mg 1 tab				
MedXN Output:				
ID6_2015-12-28_note1.txt|Lamictal::219::227|196502|80 mcg::254::260`4.5 mcg::261::268`10 mg::350::355`800 mg::399::405`800 mg::456::462|1::229::230`2::299::300`1::463::464|tablet::231::237`aerosol::283::290`puffs::301::306`tab::465::468|po::238::240`mouth::359::364`mouth::409::414|twice a day::324::335`once daily prn::365::379`every 6-8 hours as needed::415::440||				
Gold Standard I:				
ID6_2015-12-28_note1.txt	Lamictal	80 mcg	2	twice a day
ID6_2015-12-28_note1.txt	Lamictal		1	
ID6_2015-12-28_note1.txt	Lamictal	80 mcg	1	
ID6_2015-12-28_note1.txt	Lamictal	4.5 mcg	2	twice a day
ID6_2015-12-28_note1.txt	Lamictal	10 mg		once daily prn
ID6_2015-12-28_note1.txt	Lamictal	800 mg		every 6-8 hours as needed
ID6_2015-12-28_note1.txt	Lamictal	800 mg	1	
Part I Output:				
ID6_2015-12-28_note1.txt	Lamictal	80 mcg	1	
ID6_2015-12-28_note1.txt	Lamictal	4.5 mcg	1	
ID6_2015-12-28_note1.txt	Lamictal	10 mg	2	twice a day

ID6_2015-12-28_note1.txt	Lamictal	10 mg	2	once daily prn
ID6_2015-12-28_note1.txt	Lamictal	800 mg		every 6-8 hours as needed
ID6_2015-12-28_note1.txt	Lamictal	800 mg	1	

Table 5: Example of gap size causing difference in pairing between the Gold Standard I and Part I

MedXN Output:				
ID7_2012-07-03_note1.txt|lamotrigine::2076::2087|28439|100 mg::2088::2094|2::2163::2164`2::2209::2210|tablet::2095::2101`tabs::2211::2215|oral::2113::2117|bid::2118::2121`daily::2165::2170				
Gap Size:				
100mg::2088 Gap of 30 bid::2118 Gap of 45 2::2163 Gap of 2 daily::2165 Gap of 44 2::2209				
Gold Standard I:				
ID7_2012-07-03_note1.txt	lamotrigine	100 mg	2	bid
ID7_2012-07-03_note1.txt	lamotrigine	100 mg	2	daily
Part I Output:				
ID7_2012-07-03_note1.txt	lamotrigine	100 mg		bid
ID7_2012-07-03_note1.txt	lamotrigine	100 mg	2	daily
ID7_2012-07-03_note1.txt	lamotrigine	100 mg	2	

Figure 1: Diagram of Part I

Part I consists of five main steps. Case 1 illustrates steps 1 and 2, converting the NLP raw output to a standardized form and removing invalid drug names. Case 2 and Case 3 show how clusters are formed in step 3 using a gap size of 32. Case 2 illustrates a simple example of forming combinations of entities, and Case 3 shows a more complicated example. Case 3 also shows the final step of removing bad combinations.

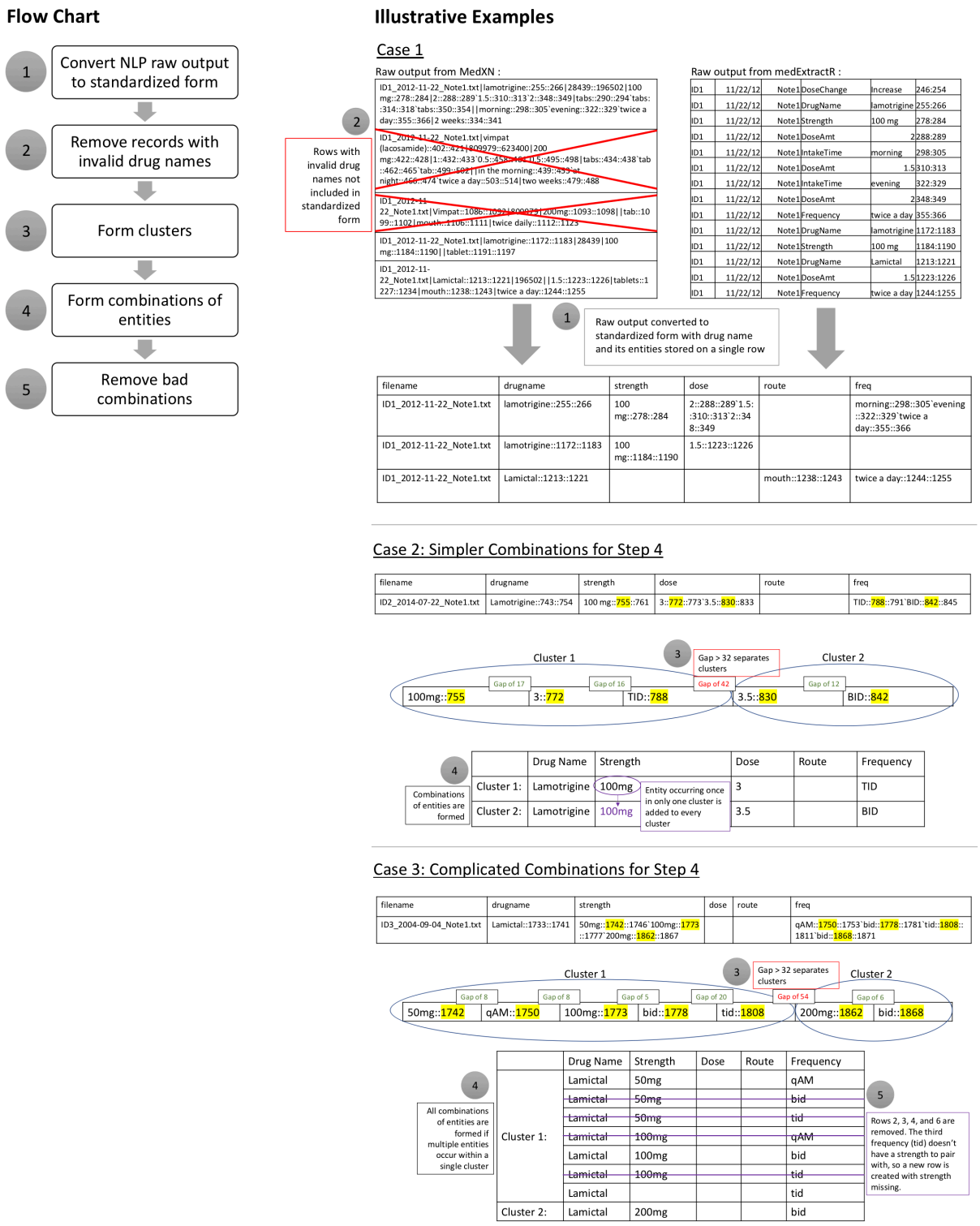
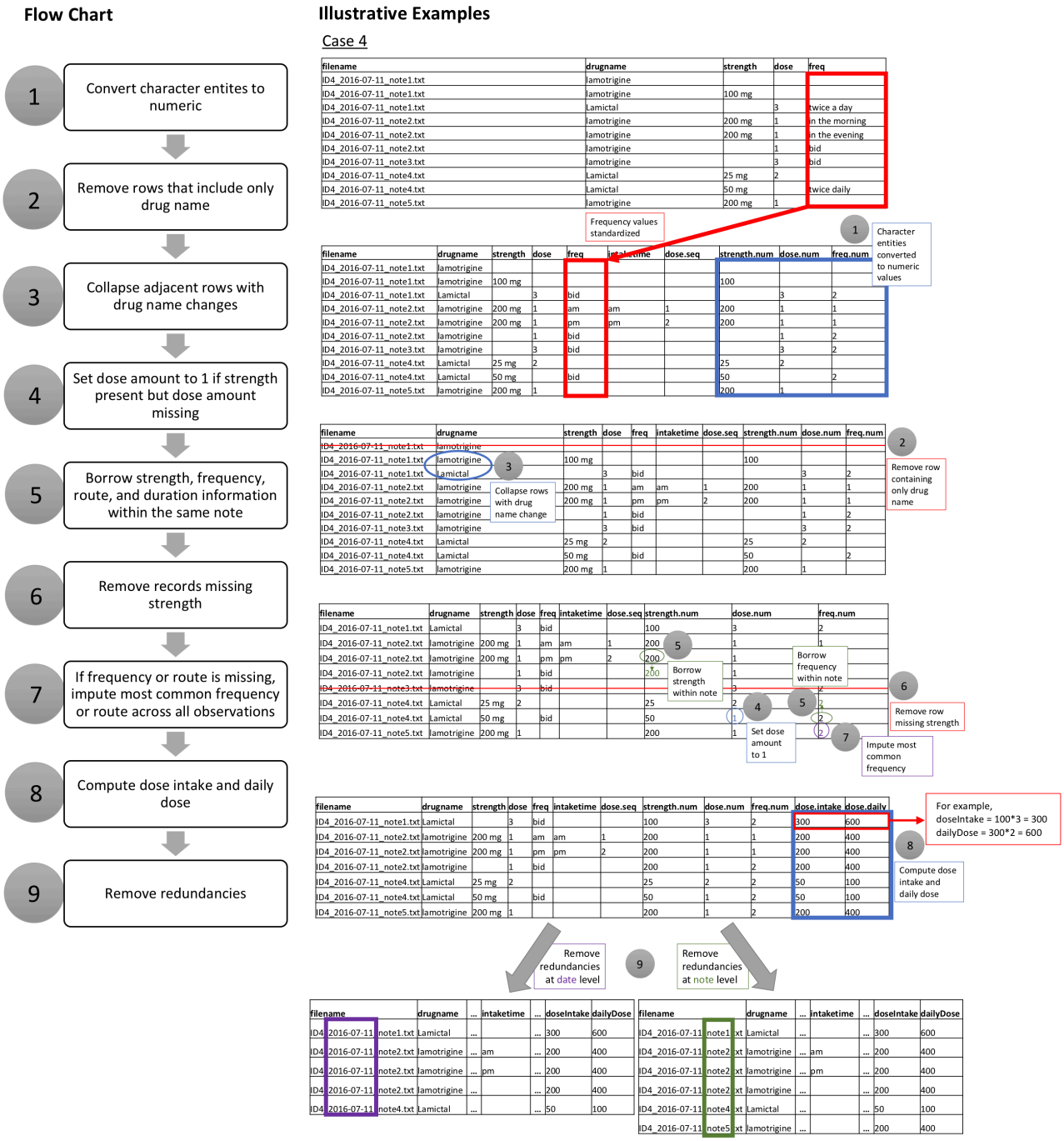


Figure 2: Diagram of Part II

Part II consists of nine main steps. Case 4 illustrates examples of each of these steps. The final output of Part II is two datasets, one with redundancies removed at the date level and one with redundancies removed at the note level.