AMIA Annu Symp Proc 2011:1117-1126.

An Approximate Matching Method for Clinical Drug Names

Lee Peters[1], M.S., Joan E. Kapusnik-Uner[2], Pharm.D., Thang Nguyen, M.S.[1], Olivier Bodenreider[1], M.D., PhD
[1] National Library of Medicine, National Institutes of Health, Bethesda, Maryland, USA
[2] First DataBank, South San Francisco, California, USA

{lpeters|nguyentd4|obodenreider}@mail.nih.gov
Joan_Kapusnik@firstdatabank.com

Abstract

Objective: *To develop an approximate matching method for finding the closest drug names within existing RxNorm content for drug name variants found in local drug formularies.* ***Methods:*** *We used a drug-centric algorithm to determine the closest strings between the RxNorm data set and local variants which failed the exact and normalized string matching searches. Aggressive measures such as token splitting, drug name expansion and spelling correction are used to try and resolve drug names. The algorithm is evaluated against three sets containing a total of 17,164 drug name variants.* ***Results:*** *Mapping of the local variant drug names to the targeted concept descriptions ranged from 83.8% to 92.8% in three test sets. The algorithm identified the appropriate RxNorm concepts as the top candidate in 76.8%, 67.9% and 84.8% of the cases in the three test sets and among the top three candidates in 90-96% of the cases.* ***Conclusion:*** *Using a drug-centric token matching approach with aggressive measures to resolve unknown names provides effective mappings to clinical drug names and has the potential of facilitating the work of drug terminology experts in mapping local formularies to reference terminologies.*

Introduction

Naming conventions used in dictionaries and thesauri likely utilize strict editorial policy and control for terms presented. Medication formulary management systems also utilize local editorial policy and data structure formatting controls for terms and attributes presented. Localized policies make perfect sense as they can orient content consistently towards the intended local data usage, such as a formulary management for drug purchasing versus formulary management for computerized prescriber order entry (CPOE) drug ordering applications. Data processing systems that attempt to manage, aggregate or process across multiple drug formularies are posed with challenges because of terms editorial policies and data structure localizations. Thus, when attempting to produce drug concept or term mappings, we are posed with the same difficulties because of the presence of local variants in "drug names". Such mapping is often handled manually and is time consuming, produce mappings that are likely optimized for only use in a single direction and for a single use case. The mapping of drug names across drug vocabulary standards is greatly facilitated by the existence of specialized terminology integration systems such as RxNorm (described in more detail later), which encompasses many drug vocabularies containing many drug variants. Improving the capabilities of mapping tools will facilitate the creation of programmatically generated mappings or candidate mappings, which can be use case specific.

Development of a set of transformation rules specific to the clinical drug domain has been shown to improve mapping of clinical drugs to RxNorm without increasing the ambiguity of normalized strings[1]. However, automated mapping using these normalization techniques still provide only effective mapping in cases where the original string can be normalized into a form contained in the data set. Drug name variants which fail to be mapped into existing terms may contain unknown abbreviations, non-standard terms, extra or missing terms or even misspellings. Some examples:

ACCUPRIL 20 MG TAB TABLET	(extra word)
HYDROCHLOROT 50 MG TABLET	(unknown abbreviation)
Rantidine 15 ML Syrup Oral	(misspelled word)
BUTALBITAL/ASPIRIN/CAFFEINE ORAL 50-325-40 CAPSULE	(missing dosage units)

The purpose of this study is to develop methods to map these previously non-mapped strings to the "closest" drug strings in the RxNorm data set. We use the term "approximate matching" to denote a method for finding terms and concepts in the RxNorm data set which most closely resemble the drug name variant being mapped. To do this, we expand upon the normalization techniques already used by introducing more aggressive reformatting and abbreviation expansion for unrecognized words, as well as spelling correction. In contrast to the conservative approach used in our normalization algorithm, this approach aims to increase recall at the possible expense of precision. If effective, the approximate matching method would provide users with a limited number of highly-relevant suggestions of drug concepts for classification or mapping purposes, analogous to a spelling checker providing spelling suggestions for a misspelled word.

Background

A number of programs provide approximate or similarity matching to medical terms. The Unified Medical Language System® (UMLS®) Terminology Services Metathesaurus® Browser[6] has an approximate match search function which identifies medical concepts relating to the words in your search string. It uses a variation of MetaMap[7] to identify the concepts. MetaMap uses a noun phrase based parsing approach to rank the candidates, so a search string of `acetaminophen tablets` will have top suggested concepts containing the term `tablet`.

A group at Oregon Health & Science University has developed a Sequential Parser[8] to handle prescription strings as part of the RxSafe project. It uses a drug-centric approach to first find a drug name, and then parses the rest of the string to find other medication information. This parser is not currently publically available.

Our approach will be similar to the Sequential Parser in that a drug name (ingredient or Trade Mark brand name) must first be found before any other matching occurs. However, we will use a token matching approach rather than parsing for specific clinical components such as strength, dose form and frequency to determine the closest candidates.

RxNorm is a standardized nomenclature for medications produced and maintained by the U.S. National Library of Medicine (NLM) in cooperation with proprietary vendors[2, 3]. RxNorm concepts are linked by NLM to multiple drug identifiers for each of the commercially available drug databases within the UMLS® Metathesaurus®. In addition to integrating names from existing drug vocabularies, RxNorm creates standard names for clinical drugs. RxNorm has established a rich set of editorial guidelines (naming conventions, conversion of units, etc), which inform both the creation of standard names and the mapping of proprietary names to standard names. However, the required transformations are only partially automated and the creation of RxNorm relies heavily on the work of human editors.

National Drug Codes (NDCs) are unique 10 digit numeric identifiers provided by drug companies to the Food and Drug Administration for identification of all manufactured drugs. Two of the data sets used in this study contain an NDC for each variant. The NDCs can be mapped through RxNorm to identify the appropriate concept variant strings in the RxNorm data set. For example:

Variant: `Acetaminophen 120 MG Suppository (RE)` *NDC:* `45802073230`

Using the RxNorm API, the NDC maps to:

RxCUI: `198434` *RxNorm name:* `Acetaminophen 120 MG Rectal Suppository`

Generic Code Sequence Numbers (GCN_SEQNO) are numeric identifiers from First Databank Inc for Clinical Formulation drug concepts. One data set used in this study contains a GCN_SEQNO for each variant drug name sources from the MED Medication ID (MEDID) value set. The GCN_SEQNOs can be mapped through RxNorm to identify concept variant strings in the RxNorm data set. For example:

Variant: `acyclovir sodium 50 mg/ml IV` *GCN_SEQNO:* `38954`

Using the RxNorm API, the GCN_SEQNO maps to:

RxCUI: `313812` *RxNorm name:* `Acyclovir 50 MG/ML Injectable Solution`

Materials

Three test datasets were used in this study. The first one, referred to as the development set, was used for developing our approximate matching algorithm, which was evaluated on the second and third data sets, called the Surescripts test set and the MEDID test set respectively. Both the development set and the Surescripts test set are de-identified units of e-prescription data from Surescripts. The development set was extracted from a set of Surescripts data collected in 2006 and used in the AHRQ report in the evaluation of e-prescribing tools[5]. The Surescripts test set contained Surescripts data generated and de-identified over a three month period from the Emergency Room at Suburban Hospital in Bethesda, Maryland.

The MED Medication ID (MEDID) test data set from First DataBank is a level of abstraction within the NDDF Plus data hierarchy that represents the unique components of product or generic drug name, route of administration, dosage form, strength, and strength unit-of-measure. These descriptions are manually created, as opposed to programmatic generation, with flexibility that allows the description to include component information only when necessary to resolve ambiguity, provide clarification, or to aid in patient safety (e.g. Brand MedID "`Cymbalta 20 mg Cap`" versus Generic MedID "`duloxetine 20 mg Cap, Delayed Release`"). Each MEDID is associated to a preferred Clinical Formulation ID (or GCN_SEQNO code), which in turn has been assigned an RxCUI. In RxNorm, the corresponding branded drug (SBD) description would be `duloxetine 20 MG Enteric Coated Capsule [Cymbalta]`. From this data set, we only use the clinical drug name and the corresponding GCN_SEQNO code, not the other manually curated pieces of information (e.g. drug name).

The development set and Surescripts test set contained a National Drug Codes (NDC) assigned with each drug name variant, which was used to validate the accuracy in the approximate matching suggestions generated in this study. We used this data for identifying specific variability patterns and for testing the degree to which the approximate matching method can handle these patterns. The MEDID test set contains the Clinical Formulation IDs (GCN_SEQNOs) for identification of the variant to a clinical drug. The data sets were first filtered to contain only drug names which were not found in the RxNorm data set through exact matching search or the normalized string search featured in the RxNorm API[4]. The drug name variants were then compared with the drug names in the March 2011 version of the RxNorm data set.

Methods

Approach. The objective of the approximate matching method is to return the closest strings *relative to the identified drug(s)*. Consider what would happen if there was no preferential treatment for the drug name component.

Input string: `alprazolam 5 mg chewable tablet`

There is no clinical drug name in RxNorm that matches this variant. In terms of token matching, the closest strings having the highest token similarity would be many including:

`Lamotrigine 5 mg chewable tablet`

`Sorbitate 5 mg chewable tablet`

(… many more)

Clearly, identifying all clinical drugs whose dose form is `chewable table` is not what is desired. Therefore, the drug name(s) from the input string must be identified so that the candidate list of strings is limited to those containing the drug name.

Overview. Our approximate matching method uses a drug-centric token matching approach to evaluate the closeness of the strings. To do this the input string is first normalized into tokens. After this occurs the tokens are searched for drug ingredient or brand names followed by actions taken to resolve unknown tokens. Then the candidate strings are identified containing the drug names found in the input string. A similarity score is calculated for each candidate string. The strings are ranked and a concept rank is calculated for the targeted concept.

Each of these steps is discussed in more detail in the following paragraphs.

Normalizing the input string. Each data set drug name is normalized using the normalization process we developed[1] and used in the RxNorm API[4]. This function, referred to hereafter as *RxNormNorm* creates an array of word tokens representing the string. Since we also normalize the RxNorm data set through *RxNormNorm* as part of

the installation process for each monthly release, the normalized input string can then be compared with the normalized strings in RxNorm. A normalization example:

Variant:

METOPROLOL SUCCINATE 200MG TAB

After RxNormNorm:

200 metoprolol mg tablet

In the example, *RxNormNorm* expands **tab** into **tablet**, separates **200** from **mg**, and removes the salt modifier **succinate**. The tokens are ordered alphabetically.

Identifying the drug names. Identifying the drug names is the most important and complex part of the approximate matching method. This is because the drug name component is from our perspective the most important part of the drug term.

We first identified the drug names in RxNorm by creating a list of all strings in concepts that had an RxNorm term type of ingredient (IN), precise ingredient (PIN) or brand name (BN). Using this list, the input string tokens are checked with the drug name list for matches.

Several other actions are attempted to find drug names. Each input token is compared to the RxNorm word index, and if not found then several actions are attempted on the unknown token:

1. **Token splitting.** An aggressive reformatting approach is used for tokens containing both letters and numbers. In drug terminologies these values usually are separate entities. So the approximate matching method separates the alphabetic and numeric portions of the token. The new token array is normalized and then checked for drug names. For example:

   Variant term:

   Atripla600-200-300MG Oral

   After RxNormNorm:

   200 300 atripla600 mg oral

   After token splitting and RxNormNorm:

   200 300 600 atripla mg oral

   In the example above, the original variant contained no spaces between the drug and the dosage. After the initial *RxNormNorm*, an unknown token `atripla600` was found by the approximate matching method. The token `atripla600` was reformatted into two tokens – `atripla` and `600`. The resulting token stream after RxNorm normalization resulted in a drug match with atripla.

2. **Drug name expansion.** We try to expand the token into a full drug name. Many drug names variants contain shortened or abbreviated forms of drug names. An auto completion-like approach is used to attempt an expansion of the unknown name to a full form of a drug name. For shortened forms that contain more than one possibility no expansion is done. An example of drug name expansion:

   Variant term:

   CHLORZOXAZON 500MG TAB

   After RxNorm normalization:

   500 chlorzoxazon mg tablet

   After drug name expansion:

   500 chlorzoxazone mg tablet

3. **Spelling correction.** If unknown tokens still exist after the previous two actions, then spelling correction is attempted. The spelling algorithm is the same as used in the RxNorm API for spelling purposes, which

returns only drug name suggestions. The unknown token must have a minimum length of five characters for spelling correction to be tried. If multiple spelling suggestions are returned only the top candidate is used. A spelling correction example:

Variant term:

CIPROFLOXACN 500MG TAB ####

After RxNorm normalization:

500 ciprofloxacn mg tablet

After spelling correction:

500 ciprofloxacin mg tablet

Identifying candidate strings containing the drug name. Once a drug name has been identified, all strings in RxNorm containing the drug name are considered as candidates. If more than one drug is identified in the input string, the union of all strings containing any of the drugs is considered.

In cases where no drug has been identified through the previous measures, a **partial drug name match** is attempted. A candidate string list is created from tokens that are not associated with dosage or drug form words (such as numbers, "mg", "tablet", "oral", etc). This might occur if a multiple word drug name is underspecified. For example:

Variant term:

Penlac 8% oral solution

After removing all dosage and drug form tokens, find all drug names containing "Penlac":

Penlac Nail Lacquer
Penlac Nail Lacquer 8% Topical Solution
Penlac Nail Lacquer 80 MG/ML Topical Solution
ciclopirox 80 MG/ML [Penlac Nail Lacquer]
ciclopirox Topical Solution [Penlac Nail Lacquer]
CICLOPIROX 80 MG TOPICAL SOLUTION [PENLAC]
ciclopirox 80 MG/ML Topical Solution [Penlac Nail Lacquer]
ciclopirox 80 MILLIGRAM In 1 MILLILITER TOPICAL SOLUTION [Penlac]

`Penlac` does not fully specify the brand `Penlac Nail Lacquer`. In this case all strings in the RxNorm data set containing `Penlac` will be considered.

Determine the similarity value for each candidate string. The tokens of each candidate string are compared to the tokens of the input variant string and the Jaccard's coefficient is calculated to determine the similarity. Jaccard's coefficient was chosen over Dice's coefficient because Jaccard's coefficient penalizes a small number of shared entries more than Dice's coefficient. To compute the similarity between the test set variant and an existing RxNorm string, we use the calculation for Jaccard's Coefficient:

$$s_{AB} = \frac{|A \cap B|}{|A \cup B|}$$

Where A represents the set of tokens in the test set variant and B represents the set tokens in the data base term.

For example, consider two drug names strings:

`Viagra 100 mg blue pill`

`Viagra 100 mg oral tablet`

After tokenizing these strings, A = {"Viagra", "100", "mg", "blue", "pill"} and B = {"Viagra", "100", "mg", "oral", "tablet"}. The intersection of A and B is {"Viagra", "100", "mg"}. The union of A and B is {"Viagra", "100", "mg", "blue", "pill", "oral", "tablet"}. Jaccard's coefficient is:

$$s_{AB} = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{7} = 0.429$$

Rank the candidate strings. The top candidate strings are ranked by their Jaccard's coefficient score and the highest 20 candidates (duplicate strings are ignored) are returned.

Determine the concept rank. A concept rank is calculated for the input string based on the list of top 20 candidate strings. The concept rank indicates the position of the top ranked string in the list which matches the concept of the input string in relation to the top ranked string of other concepts. The table below shows an example of top drug terms, their scores (Jaccard's coefficient) and their RxCUI returned from approximate matching for a input variant of `CEFACLOR ER 500 MG TABLET SIVX`, which has an NDC that maps to RxCUI 309043.

Score	Concept Rank	RxCUI	Drug Term
0.86	1	349508	Cefaclor 500 MG Extended Release Tablet
0.75	2	309043	Cefaclor Monohydrate 500mg Oral tablet, extended release
0.75	-	349508	Cefaclor 500 MG Oral Tablet, Extended Release
0.75	2	844780	Cefaclor CD 500 MG Extended Release Tablet
0.67	4	284313	Cefaclor CD, 500 mg oral tablet, extended release
0.67	-	309043	cefaclor 500 MG 12 HR Extended Release Tablet
0.67	4	844650	Cefaclor 500 MG Extended Release Tablet [Ceclor CD]
0.67	-	844780	Cefaclor 500 MG Extended Release Tablet [Cefaclor CD]
0.60	6	349507	Cefaclor 375 MG Extended Release Tablet

From the table, concept 349508 has the top ranked string so its concept rank is 1. Concepts 309043 and 844780 have a concept rank of 2, concepts 284313 and 844650 have a concept rank of 4, and concept 349507 has a rank of 6. Therefore, the input string concept rank (representing concept 309043) is 2.

Results

For the development set of 5,566 drug name variants, the approximate matching method found matches to the targeted concept in 5,166 (92.8%) of the cases, with 400 variants (7.2%) unmatched. Table 1 shows the number of variants that attained each concept ranking. From the table, the algorithm identified the appropriate RxNorm concept as the top candidate in 76.8% of the cases and among the top 3 concepts in 92.9% of the matched cases.

# of drug variants	Target Concept Rank	% of matched total
3,968	1	76.8%
618	2	12.0%
213	3	4.1%
96	4	1.9%
68	5	1.3%
42	6	0.8%
43	7	0.8%
28	8	0.5%
30	9	0.6%
14	10	0.3%
46	>10	0.9%
5,166	All	100.0%

Table 1 - Variant Concept Rankings in the Development Set

For the Surescripts test set of 2,679 drug name variants, the approximate matching method found matches to the targeted concepts in 2,246 (83.8%) of the cases, with 433 (16.2%) drug name variant cases unmatched. Table 2 shows the number of drug name variants in each concept ranking. From the table, the algorithm identified the appropriate RxNorm concept as the top candidate in 67.9% of the matched cases and among the top 3 concepts in 90% of the matched cases.

# of drug variants	Target Concept Rank	% of matched total
1525	1	67.9%
358	2	15.9%
138	3	6.1%
68	4	3.0%
28	5	1.2%
27	6	1.2%
24	7	1.1%
9	8	0.4%
16	9	0.7%
30	10	0.6%
40	>10	1.8%
2,246	All	100.0%

Table 2 - Variant Concept Ranking in the Surescripts Test Set

For the MEDID test set of 10,266 drug name variants, the approximate matching method found matches to the targeted concepts in 9,293 (90.5%) cases with 973 (9.5%) cases unmatched. Table 3 shows the number of variants that attained each concept ranking. From the table, the algorithm identified the appropriate RxNorm concept as the top candidate in 84.8% of the cases and among the top 3 concepts in 96.2% of the matched cases.

# of drug variants	Target Concept Rank	% of matched total
7,876	1	84.8%
829	2	8.9%
230	3	2.5%
115	4	1.2%
71	5	0.8%
44	6	0.5%
43	7	0.5%
26	8	0.3%
12	9	0.1%
9	10	0.1%
38	>10	0.4%
9,293	All	100.0%

Table 3 - Variant Concept Ranking in the MEDID Test Set

Discussion

Findings. The approximate matching method was effective in finding the targeted concepts from the local drug name variants. Use of the individual components of the approximate matching algorithm helped resolve drug names

and contributed to the high success rate. It is useful to understand why some local variants did not match and how effective the various components were when resolving unknown tokens. These are explained below.

The characteristics of the data sets were markedly different. The development data set contained a significant number of run-on phrases that were missing spaces, whereas the Surescripts test set contained only a couple of such instances. The development set also contained a higher percentage of unknown tokens and it appeared some of these unknown tokens contained abbreviations of drug manufacturers such as PFIZ (Pfizer), ABBO (Abbott), NOVA (Novartis) and several others. For example: `ACCUPRIL 5MG TAB PFIZ`. The Surescripts test set contained many underspecified clinical terms, while the MEDID test set had very few.

Unmatched Variants. An analysis of the 195 variants in the Surescripts test set that were not matched to the targeted concept reveals several characteristics. The majority of these 195 variants shows an underspecified name for the clinical drug abstraction. Here are several of the variants which were not matched.

```
AMITRIPTYLIN

AMOX TR-POTASSIUM CLAV

METHSCOPOLAMINE BROMID

PROMETHAZINE-CODEINE SYRUP
```

A characteristic of these names is that they are missing the strength components (for example: 40 mg) that are usually associated with the clinical drug names. Only 32 of the 195 unmatched variants contained a strength in the name. Three of the four names above are also missing the dose form of the drug (for example: tablet) which is part of a clinical drug name. So underspecification of the terms resulted in most of the unmatched results in the test set.

Of those 32 unmatched terms containing a strength component, 16 contained the words "eye drops". For example: `TRUSOPT 2% EYE DROPS.` The approximate matching method found two drug names in this string – "`Trusopt`" and "`eye drops`", the latter being a synonym of "`Eye drops brand of Tetrahydrozoline`". So the candidate list of terms contained any strings containing these two names. The closest string containing Trusopt – "`Trusopt 2% ophthalmic solution`" – did not rank in the top 20 as there were many strings of the form *`drugname`* `2% eye drops` which had higher scores.

The unmatched variants in the MEDID test set were due largely to brand names that were unknown to RxNorm. Note that the GCN_SEQNOs for these terms refer to their generic equivalent concepts, which we attempt to resolve to via RxNorm "tradename_of" relationship when a branded clinical drug concept is found. Measures to resolve unknown tokens do not help in these instances.

Relative contribution of each component of the approximate matching algorithm. An analysis of the components used in the approximate matching – token splitting, drug name expansion, spelling correction and partial drug name match – are presented in the tables below. The usage of each component was recorded and then manually analyzed for its effectiveness, that is, when used whether it helped in finding a match to the targeted concept.

Table 4 shows the usage of each component and its effectiveness for the development set.

Component	Usages	Effective Usages
Token Splitting	144	116 (80.6%)
Drug Name Expansion	164	97 (59.1%)
Spelling Correction	238	18 (7.6%)
Partial Drug Name Match	125	91 (72.8%)

Table 4 – Component Performance Evaluation (Development Set)

The development set contained a number of variants where token splitting was needed and was effective in identifying the drug name. As mentioned earlier, the development set contained a number of unknown tokens which appeared to represent names of manufacturers. When these were resolved either by drug name expansion (for example: PFIZ expanded to Pfizerpen) or by spelling correction, these resolutions were not effective in finding a targeted concept. In addition many spelling correction was applied on unknown tokens which were not intended to

be drugs (for example: CAPSUL). Partial drug name match was effective and utilized in many cases where a brand name was underspecified (for example: `Armour` instead of `Armour Thyroid`).

Table 5 shows the usage of each component and its effectiveness for the Surescripts test set.

Component	Usages	Effective Usages
Token Splitting	2	1 (50%)
Drug Name Expansion	111	87 (78.4%)
Spelling Correction	59	11 (18.6%)
Partial Drug Name Match	30	20 (66.7%)

Table 5 – Component Performance Evaluation (Surescripts Test Set)

Drug name expansion was the most used and most effective measure used in the Surescripts Test Set as more than 10% of all the variants contained a drug abbreviation. Spelling correction was the least effective, as most changes occurred on tokens that were not drug names.

Table 6 shows the usage of each component and its effectiveness for the MEDID test set.

Component	Usages	Effective Usages
Token Splitting	51	37 (72.5%)
Drug Name Expansion	58	36 (62.1%)
Spelling Correction	183	13 (7.1%)
Partial Drug Name Match	701	503 (71.8%)

Table 6 – Component Performance Evaluation (MEDID Test Set)

The majority of component uses in the MEDID test set involved partial drug name match. This test set contained a large number of brand names, and many were underspecified. For example, for the variant `Capzasin 0.15 % Topical Liquid`, RxNorm recognizes four brand names containing Capzasin: `Capzasin-p`, `Capzasin Quick Relief`, `Capzasin-hp` and `Capzasin-hp Arthritis Formula`. Since `Capzasin` doesn't match any of these names, the partial drug name match is used to allow any strings containing `Capzasin` to be candidates for matching. Spelling correction was largely ineffective because most cases involved trying to spell correct unknown brand names. The correction generated in many cases was another brand name, which did not match what was intended.

Limitations of the algorithm. Several limitations to the algorithm are described below.

Abbreviation processing. Abbreviations are handled at several levels in the approximate matching algorithm. The first level is handled by *RxNormNorm*, where a table of abbreviations is used to expand drug terms into their full names. For example, *tab* is expanded to *tablet*, *cap* to *capsule*, and *ASA* to *aspirin*. The abbreviations table is derived from abbreviations contained in the RxNorm data set. However, a complete search for all the abbreviated terms in the RxNorm data set has not been done, so abbreviations are missing from the abbreviation table used by *RxNormNorm*. The effect of this on the approximate matching algorithm will be to limit the candidate strings used if an unidentified abbreviation is used in the input variant string. For example, suppose there exists a variant name in RxNorm that contains the word `ASPRN` (meaning Aspirin) but that abbreviation is not in the *RxNormNorm* abbreviations table. If the input string for approximate matching is "`ASPRN 81 mg tablet`", then the approximate matching algorithm will find ASPRN in RxNorm, and drug name expansion or spelling correction are not triggered. Partial drug name match will limit the candidate set to the one instance where ASPRN exists in the database. So in the future better results should occur if a more thorough identification of abbreviations for RxNorm drug names is performed and reflected in the *RxNormNorm* abbreviations table.

For drug name expansion in the approximate matching algorithm, ambiguous abbreviations are not expanded. For example *albut* could be expanded to either *albutein* or *albuterol*, so it is not expanded. One possible solution is to modify the approximate matching to include candidate drug strings from all possible expansions of an abbreviated form in the ambiguous cases.

Spelling Correction. The approximate matching algorithm uses spelling correction to resolve unknown tokens to drug names. The spelling correction is only effective a small percentage of the time, partly because many of the unknown tokens are not drug names. For example, the unknown token *capsul* has a spelling correction of *capsin* when it was really a shortened form of capsule. One possible solution is to have a local variant table of non-drug name abbreviated words such as *capsul*, *tabl* and other non standard variations of common drug dosage or forms which could be checked in the approximate matching before drug name expansion and spelling correction occur. This could reduce bad candidates resulting from bad spelling corrections or drug name expansion.

Conclusion

The approximate matching method we developed can be easily implemented and extended to other drug terminologies. It is intended to provide assistance to indexing of drug variants to established medical terms or vocabularies where a human would determine the validity of the closest matches. The approximate matching is intended to be used as an aid to human experts in the mapping of local variant drug names to reference drug terminologies. While the mapping cannot be completely automated, we showed that the aid provided by the algorithm is consistently effective across various datasets as a match is found in 84-92% of the cases and the appropriate drug concept is identified among the top 3 candidates in 90-96% of the cases when a match is found.

The work done in this study is intended to be a precursor of an approximate matching function for the RxNorm API and RxNav[9]. Our intention is to offer an approximate matching function by October 2011, in complement to the normalization function that is already available. An alternate approach using similarity processing of strings should be examined and compared with the results presented here.

Acknowledgments

This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM).

References

1. Peters L, Kapusnik-Uner JE, Bodenreider O. Methods for managing variation in clinical drug names. AMIA Annu Symp Proc 2010:638-641.
2. RxNorm: http://www.nlm.nih.gov/research/umls/rxnorm/
3. Liu S, Ma W, Moore R, Ganesan V, Nelson S. RxNorm: prescription for electronic drug information exchange. IT Professional 2005;7(5):17-23
4. RxNorm API: http://rxnav.nlm.nih.gov/RxNormAPI.html
5. Findings from the evaluation of e-prescribing pilot sites. AHRQ Publication No.07-0047-EF. April 2007. http://healthit.ahrq.gov/portal/server.pt/gateway/PTARGS_0_1248_227460_0_0_18/Findings%20From%20The%20Evaluation%20of%20E-Prescribing%20Pilot%20Sites.pdf
6. UMLS Terminology Services Metathesaurus Browser: https://uts.nlm.nih.gov/metathesaurus.html
7. Aronson AR, Lang F. An overview of MetaMap: historical perspective and recent advances. JAMIA 2010 17:229-236
8. Rhotan D, Gorman P. Sequential Parser version 1.2 Design and Implementation Specification.
9. RxNav: http://rxnav.nlm.nih.gov/