Detecting Privacy-Sensitive Events in Medical Text

Prateek Jindal Dan Roth
Carl A. Gunter
Department of Computer Science, UIUC
{jindal2, danr, cgunter}@illinois.edu

September 20, 2013

Abstract

Recent US government initiatives have led to wide adoption of Electronic Health Records (EHRs). More and more health care institutions are storing patients' data in an electronic format. This emerging practice is posing several security-related risks because electronic data can easily be shared within and across institutions. So, it is important to design robust frameworks which will protect patients' privacy. In this report, we present a method to detect security-related (particularly drug abuse) events in medical text. Several applications can use this information to make the hospital systems more secure. For example, portions of the clinical reports which contain description of critical events can be encrypted so that it can be viewed only by selected individuals.

1 Introduction

While dealing with clinical narratives, there are several privacy concerns. Clinical narratives often contain sensitive information about the patients. In a hospital system, clinical narratives need to be visible to many people so that they can perform their respective functions. Sometimes, it is also necessary to share the clinical narratives among hospital systems. It is important that the privacy of patients should be respected while sharing such information across hospital systems.

There are several types of sensitive data that are found in the clinical narratives. We categorize the sensitive data into 5 major types below:

1. Mental health and abuse in the family
2. Drug Abuse
3. HIV data
4. Genomic data; indication of genetic information in EHRs
5. Sexually transmitted diseases

However in the data that we have, we only found significant number of drug abuse cases. We didn't find sufficient number of cases for other 4 types. So, in this study, we restrict ourselves to the cases of drug abuse.

2 Drug Abuse

Wikipedia gives the following definition of drug abuse which is consistent with the definitions of drug abuse found in medical sources like MedlinePlus etc.

> *Substance abuse, also known as drug abuse, is a patterned use of a substance (drug) in which the user consumes the substance in amounts or with methods neither approved nor supervised by medical professionals. Substance abuse/drug abuse is not limited to mood-altering or psycho-active drugs. If an activity is performed using the objects against the rules and policies of the matter (as in steroids for performance enhancement in sports), it is also called substance abuse.*

3 Task Description

In this chapter, following 3 things will be addressed:

1. To identify the concepts related to drug abuse.
2. To identify the assertion status (positive or negative) of concepts.
3. To identify whether the concept belonged to the patient.

4 Datasets for Experiments

For our experiments, we used the clinical narratives made available by i2b2 team as part of 2011 i2b2/VA coreference challenge. These clinical narratives came from 2 institutions: (a) Partners HealthCare, Boston and (b) Beth Israel Deaconess Medical Center.

Data was annotated by 2 annotators where one of them was a medical expert. Now, we report the results on Inter-Annotator agreement (IAA) on 10 documents. There were a total of 57 concepts related to drug abuse in the data that we selected.

For concept extraction, there was disagreement over 4 cases. So, IAA for concept extraction = 92.9%.For determining assertions, there was disagreement over 3 cases. All the cases of disagreement were related to mild alcohol usage. So, IAA for assertion detection = 94.7%. Finally, we decided that all cases of drug abuse (whether mild or strong) should be annotated to be positive. For determining experiencer of the drug abuse event, there was total agreement. So, IAA = 100.0%.

Since we had very limited data, we decided to use semi-supervised methods for finding drug-abuse events. We reserved all the annotated data for testing.

5 Method Description

In the next few subsections, we describe the methodology that we used.

5.1 Concept Identification

Concept identification was done using dictionary lookup. We compiled a list of commonly used substances used for drug abuse from web sources. Next, we obtained all the phrases appearing in the clinical narratives using a shallow parser. All those phrases which contained any term located in the drug-abuse dictionary were considered to be drug-abuse events.

5.2 Assertion Status

We adapted 3 state-of-the-art expert systems to find the assertion status of the concepts. Below, we describe these three systems in more detail:

5.2.1 Callkit

This is an implementation of the ConText algorithm [1, 2] by Imre Solti. It first of all identifies the trigger words for the negation. Consider the following sentence as an example:

The patient denies any IV drug use but did describe cocaine use for last 2 months.

In the above sentence, *'denies'* is the trigger word for negation. It is important to note that the algorithm differentiates between pseudo-triggers (like 'no increase', 'not cause' etc.) and the actual trigger words.

After determining the triggers, the algorithm determines the scope of the trigger words. The scope of a trigger word generally starts from the word to the right of the trigger and extends till the end of the sentence. But certain termination words (like 'but' in the above example) can cause the scope of a trigger to end early. Also, for certain triggers, the scope lies to the left of the trigger instead of the right. For example, consider the following sentence:

Lung injury was ruled out by the MRI exam.

In the above sentence, the scope of *'was ruled out'* is *'Lung injury'*.

Then if a concept falls within the scope of some trigger word for negation, its scope is changed to negative.

5.2.2 UtahConText

This has similar implementation as that of Callkit. However, it uses slightly different lists of trigger words.

5.2.3 MSRA

Just like ConText algorithm, it also keeps a list of trigger words and identifies the scope of trigger words. However, it addresses the issue that there may be multiple trigger words whose scope may span the concept. To resolve such a thing, it maintains a score for all possible categories. Whenever the concept falls under the scope of some trigger word, it updates the score of the corresponding category. Finally, the category with the maximum score wins. The following scoring formula was used in our implementation. It should be noted that the scoring formula depends on the distance because it

P	97.9
R	82.5
F1	89.5

Table 1: This table shows the performance of concept extraction for drug-abuse concepts.

	Negation	Experiencer
Callkit	97.9	93.6
Utah	100.0	95.7
MSRA	100.0	95.7

Table 2: This table compares the performance of three systems for negation and experiencer detection for drug-abuse concepts.

is intuitive that when a concept is close to the trigger word, then it is more likely that the trigger word is associated with the concept.

$$x_{category} = \begin{cases} 1 & \text{if } d - w \leq 0 \\ 0.8 & \text{if } d - w = 1 \\ 0.6 & \text{if } d - w = 2 \\ 0.4 & \text{if } d - w = 3 \\ \frac{1}{d-w} & \text{if } d - w \geq 4 \end{cases} \tag{1}$$

where window size was chosen to be 3.

5.3 Patient or not

All the 3 systems described above give information about the experiencer of the event as well. The mechanism used to identify the experiencer is exactly the same as described for determining the assertion.

6 Results

Table 1 shows the results for concept identification in terms of Precision, Recall and F1 scores. We see from this table that although we achieved very high precision, recall is somewhat low.

Table 2 gives the results for assertion and experiencer determination for correctly identified concepts. We find that all systems perform quite well in detecting negation and experiencer. Utah and MSRA performed the best.

7 Error Analysis

We can note from the above section on results that our system has a somewhat lower recall for concept identification. This is because of the reason that the list of substances used for drug-abuse that we generated was not comprehensive enough. In our list, we included the commonly used drug-abuse substances. However, the error analysis showed that several other substances are also used for drug-abuse. Some of the concepts that we missed include the following: *codeine*, *morphine sulphate*, *etoh*, *IVDU*, *drug use*, *drunk heavily*, *illicit substances* and *pack-year history*.

For negation and experiencer detection, we made mistakes on cases which are particularly difficult. For example, consider the following sentence:

Patient's primary care provider was called to discuss outpatient plans to help the patient stop smoking .

In the above sentence, the phrase '*patient stop smoking*' can mislead the system to predict a negated event. However, when we see the overall context, we can see that the patient is still continuing with his/her smoking habit. Next, consider the following sentence:

He works as a counselor at an alcohol and drug treatment facility for teenagers .

In the above sentence, the word '*alcohol*' can mislead the system to predict a positive drug-abuse event. However, there is no drug-abuse (either positive or negative) being reported here at all.

8 Medical Set Expansion

In Section 7, we saw that our system has somewhat low recall for concept identification. For concept identification, we have very limited annotated data. This prevents us from developing a supervised learning approach for concept identification.

8.1 Semi-Supervised Methods for Concept Identification

In the literature, several semi-supervised methods have been proposed for concept identification. The essential underlying principle behind these semi-supervised methods is that of bootstrapping. In bootstrapping, the input consists of a few examples (also called seeds) of the concept type which we are interested in. Then the system tries to grow the seed set by finding concepts which are similar to the seeds. Distributional context of the concepts generally provides a good way to test the similarity of any two concepts. Bootstrapping approach terminates when the system is unable to grow the seed set further.

For bootstrapping approach to be successful, there should be a lot of instances of the concepts which we are interested in. If this is not the case, then the distributional context of the concepts would be very sparse and thus, insufficient for computing the similarity between two mentions. This is exactly the problem that we face in the datasets that we are experimenting with. These datasets have very few instances of "drug abuse" events, thus, limiting the usefulness of bootstrapping approach.

8.2 Active Learning Solution for Concept Identification

Since our datasets have only few instances of relevant concepts, we need to provide some extra level of supervision to our concept identification system. We rely on active learning methods to provide this extra level of supervision. In an active learning based solution, the system asks some questions to the user. The answers provided by the user are used by the system to learn a model for identifying relevant concepts. A good active learning system should ask minimal number of questions from the user.

Moreover, since we lack a good distributional context of the relevant concepts, we use the tree positions of the concepts in a medical encyclopedia named SNOMED CT to find the similarity between mentions.

8.3 Using SNOMED CT for Medical Set Expansion

Using SNOMED CT, we build a detailed descriptor of every concept. Every concept can appear at multiple places in SNOMED CT. We define the descriptor of a concept to be simply the parents of the concept upto 5 higher levels. We explain it below with the help of an example. Let us consider the concept "cocaine". The descriptor of this concept is shown in Table 3. At level 0, two SNOMED CT concepts corresponding to "cocaine" are shown.

Level	Concepts	
Level 0	Cocaine,	Cocaine measurement
Level 1	Drug measurement, **Psychostimulant,**	Tropane alkaloid, Ester type local anesthetic
Level 2	Azabicyclo compound, Alkaloid, Measurement of substance, Heterocyclic compound, Psychotherapeutic agent	Local anesthetic, Ester, Stimulant, Tropane alkaloid,
Level 3	CNS drug, Aza compound, Heterocyclic compound, Azabicyclo compound, Drug pseudoallergen by function, Tropane alkaloid,	**Psychoactive substance,** Anesthetic, Measurement, Organic compound, Alkaloid, Psychotherapeutic agent
Level 4	CNS drug, Aza compound, Techniques, Heterocyclic compound, Organic compound, Alkaloid, Psychotherapeutic agent, General drug type	**Psychoactive substance,** Evaluation procedure, Chemical categorized structurally, Azabicyclo compound, Drug pseudoallergen, Tropane alkaloid, Substance categorized functionally,

Table 3: This table shows the descriptor for concept "cocaine".

Concepts at any level $i+1$ are basically the parents of concepts at level i. It is normal for some of the concepts to repeat at later levels. These descriptors were made by a simple breadth-first search on the SNOMED CT graph starting from the concept under consideration.

8.4 User Involvement

In this subsection, we describe how the user contributes to the learning of a model for concept identification. To begin with, input to the system consists of a few seeds. Let us represent this seed set by $\mathcal{S}$. Let s_i denote the i^{th} element of seed set. For finding the substances which are potentially used for drug abuse, the input can be the following: "cocaine", "marijuana", "alcohol". Then the system computes the descriptors of each of the concepts and then merges those descriptors into a single descriptor. Let us assume that for concept x, parents at level i are denoted by the set $\mathcal{L}_i(x)$. Then the levels of the overall descriptor are defined by the following equation:

$$\mathcal{L}_i(\mathcal{S}) = \bigcup_{j=1}^{|\mathcal{S}|} \mathcal{L}_i(s_j) \qquad \forall i \tag{2}$$

After some preprocessing (like removing overly general concepts), the descriptor is shown to the user. Then the user is supposed to identify one or more most appropriate SNOMED CT concepts from the descriptor. User response is recorded into a list. Let us call this list as $MedRep(\mathcal{S})$. No further input from user is now required.

8.5 Computing the Score of a Concept

In this subsection, we describe how to compute the similarity of any given SNOMED CT concept to the seed set, $\mathcal{S}$, provided by the user. Let us denote the given SNOMED CT concept by the variable x. Also, assume that for concept x, parents at level i are denoted by the set $\mathcal{L}_i(x)$. Then the similarity, $sim(x, \mathcal{S})$, of the concept x to the seed set $\mathcal{S}$ is defined by the following equation:

$$sim(x, \mathcal{S}) = \left| \left(\bigcup_{i=1}^{4} \mathcal{L}_i(x) \right) \bigcap MedRep(\mathcal{S}) \right| \tag{3}$$

In other words, similarity of a concept to the seed set is the number of *unique* SNOMED CT concepts in the descriptor of the concept that also appear in the representative model of the seed set given to the system.

8.6 Performing Concept Identification

After receiving the user input, the system proceeds to find the relevant concepts from the provided dataset. Relevant concepts are found using the following steps:

1. First of all, we use a chunker to find all the NPs (noun phrases) in the given document.
2. Each of the noun phrases found in Step 1 is mapped to SNOMED CT concepts using a biomedical engine (MetaMap).

Algorithm 1: MedicalSetExpansion

```
Input  : $\mathcal{S}$ (Seed Set), $\mathcal{D}$ (Document Set)
Output: $\mathcal{R}_{\mathcal{D}}(\mathcal{S})$ (Ranked List of concepts)
begin
    for every seed $s \in \mathcal{S}$ do
        Compute the descriptor of $s$ using Breadth First Search on
        SNOMED CT graph
    Compute the overall descriptor of $\mathcal{S}$ by merging the individual
    descriptors according to Equation (2)
    Display the overall descriptor to user after some pre-processing
    Record user response in $MedRep(\mathcal{S})$
    for each noun phrase $x$ in $\mathcal{D}$ do
        Compute $sim(x, \mathcal{S})$ according to Equation (3)
    $\mathcal{R}_{\mathcal{D}}(\mathcal{S}) \longleftarrow$ List of NPs sorted by similarity (descending order)
```

3. Then we compute the score of each NP as described in previous subsection (§8.5).
4. Finally, the noun phrases are displayed to the user in decreasing order of score.

Algorithm 1 explains the overall algorithm for medical set expansion.

9 Focussing on Drug Abuse Events

Using the concept recognition technique described in §8.6, it is possible to build a recognizer for any concept type that we may be interested in. For example, one may build a recognizer for finding out the mentions of heart problems. Other examples of recognizers include lung problems, kidney problems, pain-killers, closed surgeries, drug abuse events, sex-related matters, genomic data etc.

In this section, we will focus on the recognizer for drug abuse events. In §5.1, we described a recognizer for drug abuse events based on dictionary lookup. §A gives a list of popular drugs that are often used for abuse. This list was compiled from these websites: Wikipedia[1], SAMHSA[2], MedlinePlus[3]

[1] http://en.wikipedia.org/wiki/Substance_abuse

[2] http://www.samhsa.gov/

[3] http://www.nlm.nih.gov/medlineplus/drugabuse.html

and WebMD[4].

In §8.6, we described a yet another technique of concept recognition using medical set expansion. In that technique, model for concept identification consists of a list (called as $MedRep(\mathcal{S})$) which basically contains the representatives of the desired concept type in a medical encyclopedia (namely SNOMED CT). For the concept type "drugs used for substance abuse", $MedRep(\mathcal{S})$ contains the following elements:

1. Psychoactive substance
2. Alcoholic Beverage
3. Central Depressant
4. Alcohol agent
5. Alcohol products
6. Substance of abuse
7. Cannabis
8. Hallucinogen
9. Cannabinoid
10. Nicotiana
11. Tobacco
12. Tobacco smoking behavior
13. Tobacco use and exposure
14. Psychotherapeutic agent
15. Psychostimulant
16. Opiate
17. Morphine Derivative
18. Analgesic
19. Anesthetic
20. Drugs used to treat addiction
21. Carboxylic acid and/or salt
22. Barbiturate
23. Centrally acting muscle relaxant

[4] http://www.webmd.com/mental-health/substance-abuse

P	95.1
R	89.3
F1	92.1

Table 4: This table shows the performance of concept extraction for drug-abuse concepts.

24. Centrally acting hypotensive agent
25. Cardiovascular agent
26. Sympathomimetic agent
27. Aralkylamine
28. Inhaled Drug Administration
29. Hypnotics
30. Anxiolytic, sedative AND/OR hypnotic

The above list was obtained using just a few seed words like cocaine, hashish, beer, wine, cannabis, smoking etc.

9.1 Results

Table 4 shows the results for concept identification for the dataset described in §4. We see from this table that the recall improved from 82.5 to 89.3 whereas the precision dropped a little. Overall, the F1 score increased from 89.5 to 92.1.

To further test the effectiveness of our system in identifying the substances used for drug abuse, we prepared a dataset using medical forums where people discuss issues related to addiction with drugs. The dataset contained a total of 135 distinct substances that can be used for drug abuse. Out of these 135 substances, our system could correctly identify 55 substances. Thus, we achieved a recall of 40.7.

10 Error Analysis

The above results indicate that our system still misses many drugs that are used for abuse. §B gives a list of drugs that were missed by our system. Below we identify the main reasons for missing such drugs:

1. One primary reason for the low recall was that SNOMED CT does not always have the trademark names for the drugs. For example, *Lorazepam* is a drug that can potentially be abused. Its tradename is *Ativan*. Although, SNOMED CT has an entry for *Lorazepam*, it does not have an entry for *Ativan*. Similar thing happened with the concepts *Percocet*, *Vicodin*, *Darvocet*, *Ritalin* and *Lorcet* which were tradenames for *oxycodone*, *hydrocodone*, *propoxyphene*, *methylphenidate* and *hydrocodone bitartrate* respectively.

2. Another reason for the low recall is that sometimes the drugs are mentioned by their street names which are not present in SNOMED CT. For example, street names for the drug *marijuana* are *ganja*, *grass*, *green*, *Mary Jane* etc. Similarly, street names for the drug *cocaine* are *candy*, *Charlie*, *toot*, *crack* etc.

3. Third reason for the low recall is that SNOMED CT sometimes doesn't have the abbreviations for the drug names. For example, it does not have the abbreviations LAAM (levacetylmethadol), PCP (phencyclidine) etc.

11 Future Work

Following are the good directions for the future work:

1. Wikipedia has a lot of medical knowledge. As discussed above in §10, a good amount of knowledge in Wikipedia is not even covered in medical encyclopedias like SNOMED CT. So, it will be a very good project to extract the medical knowledge in Wikipedia and put it in a structured database. For example, Wikipedia can tell the tradenames and common abbreviations for a lot of drugs. Following are the good sources of information in Wikipedia:

   (a) Hyperlinks in free text

   (b) Redirect Pages

   (c) Disambiguation Pages

   (d) Infoboxes

2. Like Wikipedia, there are several other sources of medical information on the web. One very good source for medical information is MedlinePlus. It will be good to extract medical information from it. There is another website, MediLexicon, which gives a lot of useful medical abbreviations.

3. Another good way to get useful medical knowledge is to send automated queries to web search engines. The top pages from the search results can then be used to glean useful medical information. It will be good to design the protocol such that the queries to the search engine are minimized because some search engines block the IP addresses which send too many queries.

12 Related Work

The task of *set-expansion* has been addressed in several works. We report here the most significant efforts towards this task.

12.1 Web-based Set-Expansion systems

Google^{TM} has a proprietory system, Google Sets[5], for set-expansion. Google Sets make use of the lists identified by the Google search engine as it crawls the web. Items given as input to the Google Sets are matched up against these lists and probabilities are calculated to determine which items might be a good match for the desired concept. The lists produced by Google Sets are quite small ($\leq$ 50). Google Sets has been used for a number of purposes in the research community, including deriving features for named-entity recognition [3], and evaluation of question answering systems [4].

Another system for *set-expansion* is Boowa[6] [5, 6, 7, 8, 9]. Like Google Sets, Boowa works by finding semi-structured web pages that contain "lists" of items, and then aggregating these "lists" so that the most promising items are ranked higher. Unlike Google Sets, Boowa produces extensive lists. Boowa accepts a maximum of 3 seeds. Since Boowa tries to find all the seeds on the same web-page, its performance may go down with increasing number of seeds as is also the case with Google Sets. The KnowItAll system of Etzioni et al. [10] depends on the output of existing search engines to extract collections of facts from the Web. Etzioni et al. [10] use Pattern Learning, Subclass Extraction and List Extraction to improve KnowItAll's recall.

12.2 Set-Expansion systems for free text

For set-expansion on free-text, pattern recognition and distributional similarity have primarily been used.

[5] `http://labs.google.com/sets`

[6] `http://www.boowa.com/`

Riloff and Jones [11] used a two-level bootstrapping mechanism based on pattern recognition for set-expansion. Their algorithm starts with a few seeds from the category of interest. Then they run multiple iterations where in each iteration, they add 5 new members to the list. Since they need to make one pass over the entire corpus for every iteration, their method is quite inefficient. Moreover, their algorithm is very sensitive to the erroneous members which may get added to the list during the expansion.

Talukdar et al. [12] present a context pattern induction method for named-entity extraction. Their method automatically selects trigger words to mark the beginning of a pattern, which is then used for bootstrapping from free text. However, they focussed on very broad entity types like Location, Person and Organization whereas we are interested in finer concepts like Athletes, Actors etc. Moreover, they used hundreds of seeds for constructing the semantic lexicons. On the other hand, we give a much smaller number of seeds.

Sarmento et al. [13] present a corpus-based approach to set-expansion. For a given set of seed entities they use co-occurrence statistics taken from a text collection to define a membership function that is used to rank candidate entities for inclusion in the set. They represent entities as vectors and essentially construct a centroid of the seed-set.

Pantel et al. [14] developed a parallel implementation for computing the pairwise semantic similarity between the entities. They applied the learned similarity matrix to the task of set-expansion using the centroid-based algorithm developed by Sarmento et al. [13]. They present a large empirical study to quantify the effect of corpus size, corpus quality, seed composition and seed size on set-expansion performance.

12.3 Set-Expansion systems using Integrated approaches

Talukdar et al. [15] present a graph-based semi-supervised label propagation algorithm for acquiring open domain labeled classes and their instances from a combination of unstructured and structured text sources. Pennacchiotti and Pantel [16] present a framework called Ensemble Semantics for modeling information extraction algorithms that combine multiple sources of information and multiple extractors. Pasca and Van Durme [17] present an approach to information extraction that exploits both Web documents and query logs to acquire open-domain classes of instances, along with relevant sets of open-domain class attributes.

Ghahramani and Heller [18] illustrates a Bayesian Sets algorithm that solves a particular sub-problem of set-expansion, in which candidate sets are given, rather than a corpus of documents.

12.4 Use of Negative Examples in Set-Expansion

Thelen and Riloff [19] and Lin et al. [20] present a framework to learn several semantic classes simultaneously. In this framework, the instances which have been accepted by any one semantic class serve as negative examples for all other semantic classes. This approach is limited because it necessitates the learning of several semantic classes simultaneously. Moreover, negative examples are NOT useful if the different semantic classes are not related to one another. Winston et al. note that it is not easy to acquire good negative examples. The approach presented by us allows the use of negative examples even when there is only one semantic class. Also, we present a strategy to easily acquire good negative examples.

In this chapter, we focus on set-expansion from free text. So, we don't compare our system with the systems which use textual sources other than free text (e.g. semi-structured web pages or query logs). The works of Sarmento et al. [13] and Pantel et al. [14] are the state-of-the-art works that come closest to our approach. In our experiments, we compare the centroid-based approach employed by them with the approach developed by us.

12.5 Using Wikipedia

Recently, there has been a lot of work centered around Wikipedia. Ratinov et al. [21] analyze local and global approaches for disambiguation to Wikipedia. Yan et al. [22] present an unsupervised relation extraction method for discovering and enhancing relations in which a speciïňĄed concept in Wikipedia participates. Using respective characteristics of Wikipedia articles and Web corpus, they develop a clustering approach based on combinations of patterns: dependency patterns from dependency analysis of texts in Wikipedia, and surface patterns generated from highly redundant information related to the Web. Nguyen and Moschitti [23] extend distant supervision (DS) based on Wikipedia for Relation Extraction (RE) by considering (i) relations deïňĄned in external repositories, e.g. YAGO, and (ii) any subset of Wikipedia documents. They show that training data constituted by sentences containing pairs of named entities in target relations is enough to produce reliable supervision. Wu and Weld [24] present WOE, an open IE system which improves dramatically on TextRunnerâĂŹs [25, 26] precision and recall. The key to WOEâĂŹs performance is a novel form of self-supervised learning for open extractors âĂŤ using heuristic matches between Wikipedia infobox attribute values and corresponding sentences to construct training data.

Conclusion

In this report, we presented a study on the detection of drug abuse events in medical text. We explored different state-of-the-art techniques for determining the negation status and experiencer of drug abuse events. For finding the drug abuse concepts, we used an active learning based approach to set expansion. The medical knowledge needed in set-expansion process was obtained from SNOMED CT. We showed that our concept identification technique is able to successfully find even uncommon drugs which are used for abuse. However, since SNOMED CT does not have tradenames and street names for many concepts, a good direction for future research is to augment the current system with the knowledge from web.

Acknowledgments

We thankfully acknowledge the contribution of Tony Michalos in helping us to prepare the datasets used in the experimental work. We are also thankful to Prof. Ozlem Uzuner for making available the clinical narratives used in i2b2 NLP challenges. This research was supported by Grant HHS 90TR0003/01. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS or the US government.

A Popular Drug Abuse Substances

Following is a list of most popular substances which are used for drug abuse:

1. alcohol
2. amphetamines
3. anabolic steroids
4. barbiturates
5. beer
6. benzodiazepines (particularly alprazolam, temazepam, diazepam and clonazepam)
7. buprenorphine
8. butane
9. cannabis
10. club drugs
11. cocaine
12. depressants (sedatives)
13. ecstasy
14. GHB
15. hallucinogens
16. heroin
17. inhalants
18. ketamine
19. LSD
20. marijuana
21. mephedrone
22. methamphetamine
23. methadone
24. methaqualone
25. narcotics
26. opioids
27. pain relievers

28. pcp
29. psychotherapeutics
30. qat/khat
31. rum
32. stimulants
33. tobacco
34. tranquilizers
35. whisky
36. wine

B Concepts that We Missed

Following is a list of drug abuse substances that were not detected by our software:

1. actiq
2. adderall
3. ambien
4. amytal
5. anexsia
6. antabuse
7. ativan
8. avinza
9. biocodone
10. campral
11. concerta
12. damason-P
13. darvocet
14. darvon
15. demerol
16. depade
17. desoxyn
18. dexedrine
19. dextrostat
20. di-gesic
21. dicodid
22. dilaudid
23. duodin
24. duragesic
25. duramorph
26. fioricet
27. fiorinal

28. halcion
29. hycodan
30. hydrococet
31. kadian
32. kapanol
33. klonopin
34. LAAM
35. librium
36. lorcet
37. lortab
38. luminal
39. ms contin
40. msir
41. methadrine
42. mushrooms
43. nembutal
44. norco
45. oramorph
46. orlaam
47. PCP
48. palladone
49. panacet
50. percocet
51. percodan
52. quaalude
53. revia
54. ritalin
55. rohypnol
56. roxanol
57. roxicodone

58. ryzolt
59. seconal
60. soma
61. speed
62. steroids
63. stilnox
64. sublimaze
65. suboxone
66. subutex
67. symtan
68. temesta
69. tramal
70. tussionex
71. tylox
72. ultram
73. valium
74. vicodin
75. vicoprofen
76. vivitrol
77. xanax
78. xodol
79. zydone

References

[1] H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman, "Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports," *Journal of biomedical informatics*, vol. 42, no. 5, pp. 839–851, 2009.

[2] W. Chapman, W. Bridewell, P. Hanbury, G. Cooper, and B. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *Journal of biomedical informatics*, vol. 34, no. 5, pp. 301–310, 2001.

[3] B. Settles, "Biomedical named entity recognition using conditional random fields and rich feature sets," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. Association for Computational Linguistics, 2004, pp. 104–107.

[4] J. Prager, "Question answering using constraint satisfaction," in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04*. Citeseer, 2004.

[5] R. Wang and W. Cohen, "Language-independent set expansion of named entities using the web," in *ICDM*. IEEE Computer Society, 2007, pp. 342–350.

[6] R. Wang and W. Cohen, "Iterative set expansion of named entities using the web," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 1091–1096.

[7] R. Wang and W. Cohen, "Character-level analysis of semi-structured documents for set expansion," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 2009, pp. 1503–1512.

[8] R. Wang, N. Schlaefer, W. Cohen, and E. Nyberg, "Automatic set expansion for list question answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 947–954.

[9] R. Wang and W. Cohen, "Automatic set instance extraction using the web," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 441–449.

[10] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial Intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[11] E. Riloff and R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," in *Proceedings of the National Conference on Artificial Intelligence*. JOHN WILEY & SONS LTD, 1999, pp. 474–479.

[12] P. Talukdar, T. Brants, M. Liberman, and F. Pereira, "A context pattern induction method for named entity extraction," in *Proceedings of the Tenth Conference on CoNLL*. ACL, 2006, pp. 141–148.

[13] L. Sarmento, V. Jijkuon, M. de Rijke, and E. Oliveira, "More like these: growing entity classes from seeds," in *Proceedings of the sixteenth ACM conference on CIKM*. ACM, 2007, pp. 959–962.

[14] P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas, "Web-scale distributional similarity and entity set expansion," in *Proceedings of the 2009 Conference on EMNLP*. ACL, 2009, pp. 938–947.

[15] P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira, "Weakly-supervised acquisition of labeled class instances using graph random walks," in *Proceedings of the Conference on EMNLP*. ACL, 2008, pp. 582–590.

[16] M. Pennacchiotti and P. Pantel, "Entity extraction via ensemble semantics," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 238–247.

[17] M. Pasca and B. Van Durme, "Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs," in *Proceedings of the 46th Annual Meeting of the ACL (ACL-08)*. Citeseer, 2008, pp. 19–27.

[18] Z. Ghahramani and K. Heller, "Bayesian sets," *Advances in Neural Information Processing Systems*, vol. 18, p. 435, 2006.

[19] M. Thelen and E. Riloff, "A bootstrapping method for learning semantic lexicons using extraction pattern contexts," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 214–221.

[20] W. Lin, R. Yangarber, and R. Grishman, "Bootstrapped learning of semantic classes from positive and negative examples," in *Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, vol. 1, no. 4, 2003, p. 21.

[21] L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia." in *ACL*, vol. 11, 2011, pp. 1375–1384.

[22] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, and M. Ishizuka, "Unsupervised relation extraction by mining wikipedia texts using information from the web," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1021–1029.

[23] T.-V. T. Nguyen and A. Moschitti, "End-to-end relation extraction using distant supervision from external semantic repositories." in *ACL (Short Papers)*, 2011, pp. 277–282.

[24] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 118–127.

[25] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2007, pp. 25–26.

[26] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web." in *IJCAI*, vol. 7, 2007, pp. 2670–2676.