# Extracting Medication Information from Discharge Summaries

**Scott Halgrim, Fei Xia, Imre Solti, Eithon Cadag**
University of Washington
PO Box 543450
Seattle, WA 98195, USA
{captnpi,fxia,solti,ecadag}@uw.edu

**Özlem Uzuner**
University of Albany, SUNY
135 Western Ave
Albany, NY 12222, USA
ouzuner@uamail.albany.edu

## Abstract

*Extracting medication information from clinical records has many potential applications and was the focus of the i2b2 challenge in 2009. We present a hybrid system, comprised of machine learning and rule-based modules, for medication information extraction. With only a handful of template-filling rules, the system's core is a cascade of statistical classifiers for field detection. It achieved good performance that was comparable to the top systems in the i2b2 challenge, demonstrating that a heavily statistical approach can perform as well or better than systems with many sophisticated rules. The system can easily incorporate additional resources such as medication name lists to further improve performance.*

## 1 Introduction

Narrative clinical records store patient medical information, and extracting this information from these narratives supports data management and enables many applications (Levin et al., 2007). Informatics for Integrating the Biology and the Bedside (*i2b2*) is an NIH-funded National Center for Biomedical Computing based at Partners HealthCare System, and it has organized annual NLP shared tasks and challenges since 2006 (https://www.i2b2.org/). The Third i2b2 Workshop on NLP Challenges for Clinical Records in 2009 studied the extraction of medication information from hospital discharge summaries (https://www.i2b2.org/NLP/Medication/), a task we refer to as *the i2b2 challenge* in this paper.

In the past decade, there has been extensive research on information extraction in both the general and biomedical domains (Wellner et al., 2004; Grenager et al., 2005; Poon and Domingos, 2007; Meystre et al, 2008; Rozenfeld and Feldman, 2008). Interestingly, despite the recent prevalence of statistical approaches in most NLP tasks (including information extraction), most of the systems developed for the i2b2 challenge were rule-based. In this paper we present our hybrid system, whose core is a cascade of statistical classifiers that identify medication fields such as medication names and dosages. The fields are then assembled to form medication entries. While our system did not participate in the i2b2 challenge (as we were part of the organizing team), it achieved good results that matched the top i2b2 systems.

## 2 The i2b2 Challenge

This section provides a brief introduction to the i2b2 challenge.

### 2.1 The task

The i2b2 challenge studied the automatic extraction of information corresponding to the following *fields* from hospital discharge summaries (Uzuner, et al., 2010a): names of medications (*m*) taken by the patient, dosages (*do*), modes (*mo*), frequencies (*f*), durations (*du*), and reasons (*r*) for taking these medications. We refer to the medication field as the *name* field and the other five fields as the *non-name* fields. All non-name fields correspond to some name field mention; if they were specified within a two-line window of that name mention, the i2b2 challenge required such fields to be linked to the name field to form an *entry*. For each entry, a system must determine whether the entry appeared in a list of medications or in narrative text. Table 1 shows an excerpt from a discharge summary and the corresponding entries in the gold standard. The first entry appears in narrative text, and the second in a list of medication information.

| **Excerpt of Discharge Summary** |
|---|
| 55 the patient noted that he had a recurrence of this<br>56 vague chest discomfort as he was sitting and<br>57 talking to friends. He took a sublingual<br>58 Nitroglycerin without relief.<br>...<br>65 Flomax ( Tamsulosin ) 0.4 mg, po, qd,... |
| **Gold standard**:<br>m="Nitroglycerin" 58:0 58:0 \|\|do="nm"\|\|<br>mo="sublingual" 57:6 57:6 \|\|f="nm" \|\|du="nm" \|\|<br>r="vague chest discomfort" 56:0 56:2 \|\|<br>ln="narrative"<br>...<br>m="flomax ( tamsulosin )" 65:0 65:3\|\|do="0.4 mg"<br>65:4 65:5\|\|mo="po" 65:6 65:6\|\|f="qd" 65:7<br>65:7\|\|du="nm"\|\|r="nm"\|\|ln="list" |

Table 1: A sample discharge summary excerpt and the corresponding entries in the gold standard. The fields inside an entry are separated by "||". Each field is represented by the string and its position (i.e., "line number: token number" offsets). "nm" means the field value is not mentioned for this medication name.

## 2.2 Data Sets

The i2b2 challenge used a total of 1243 discharge summaries:

- 696 of these summaries were released to participants for system development, and the i2b2 organizing team provided the gold standard annotation for 17 of them.

- Participating teams could choose to annotate more files themselves. The University of Sydney team annotated 145 out of the 696 summaries (including re-annotating 14 of the 17 files annotated by the i2b2 organizing team) and generously shared their annotations with i2b2 after the challenge for future research. We obtained and used 110 of their annotations as our training set and the remaining 35 summaries as our development set.

- The participating teams produced system outputs for 547 discharge summaries set aside for testing. After the challenge, 251 of these summaries were annotated by the challenge participants, and these 251 summaries formed the final test set (Uzuner et al., 2010b).

The sizes of the data sets used in our experiments are shown in Table 2. The training and development sets were created by the University of Sydney, and the test data is the i2b2 official challenge test set. The average number of entries and fields vary among the three sets because the summaries in the test set were chosen *randomly* from the 547 summaries, whereas the University of Sydney team annotated the *longest* summaries.

## 2.3 Additional resources

Besides the training data, the participating teams were allowed to use any additional tools and resources that they had access to, including resources not available to the public. All challenge participants used additional resources such as UMLS (www.nlm.nih.gov/research/umls/), but the exact resources used varied from team to team. Therefore, the challenge was similar to the so-called *open-track* challenge in the general NLP field, as opposed to a *closed-track* challenge that could require that all the participants use only the list of resources specified by the challenge organizers

## 2.4 Evaluation metrics

The i2b2 challenge used two sets of evaluation metrics: *horizontal* and *vertical* metrics. Horizontal metrics measured system performance at the entry level, whereas vertical metrics measured system performance at the field level. Both sets of metrics compared the system output and the gold standard at the *span* level for *exact match* and at the *token* level for *inexact match*, using precision, recall, and F-score (Uzuner et al., 2010a). The primary metric for the challenge is exact horizontal F-score, which is the metric we use to evaluate our system.

| Data Sets | # of Summaries | # of Entries | # of Fields | # of Names | # of Doses | # of Freq | # of Modes | # of Duration | # of Reason |
|---|---|---|---|---|---|---|---|---|---|
| Training set | 110 | 5970 (54.3) | 14886 (135.3) | 5684 (51.7) | 2929 (26.6) | 2740 (24.9) | 2146 (19.5) | 302 (2.7) | 1085 (9.9) |
| Dev set | 35 | 2401 (68.6) | 5988 (171.1) | 2302 (65.8) | 1163 (33.2) | 1096 (31.3) | 880 (25.1) | 111 (3.2) | 436 (12.5) |
| Test set | 251 | 8936 (35.6) | 22041 (87.8) | 8495 (33.8) | 4387 (17.5) | 3999 (15.9) | 3307 (13.2) | 511 (2.0) | 1342 (5.3) |

Table 2: The data sets used in our experiments. The numbers in parentheses are the average numbers of entries or fields per discharge summary.

## 2.5 Participating systems

Twenty teams participated in the challenge. Fifteen teams used rule-based approaches, and the rest used statistical or hybrid approaches. The performances of the top five systems are shown in Table 3. Among them, only the top system, developed by the University of Sydney, used a hybrid approach, whereas the rest were rule-based.

| Rank | Precision | Recall | F-score |
|---|---|---|---|
| 1 | 89.6 | 82.0 | 85.7 |
| 2 | 84.0 | 80.3 | 82.1 |
| 3 | 86.4 | 76.6 | 81.2 |
| 4 | 78.4 | 82.3 | 80.3 |
| 5 | 84.1 | 75.8 | 79.7 |

Table 3: The performance (exact horizontal precision/recall/F-score) of the top five i2b2 systems on the test set.

## 3 System description

We developed a hybrid system with three processing steps: (1) a preprocessing step that finds section boundaries, (2) a field detection step that identifies the six fields, and (3) a field linking step that links fields together to form entries. The second step is a statistical system, whereas the other two steps are rule-based. The second step was the main focus of this study.

### 3.1 Preprocessing

In addition to common processing steps such as part-of-speech (POS) tagging, our preprocessing step includes a section segmenter that breaks discharge summaries into sections. Discharge summaries tend to consist of sections such as 'ADMIT DIAGNOSIS', 'PAST MEDICAL HISTORY', and 'DISCHARGE MEDICATIONS'. Knowing section boundaries is important for the i2b2 challenge because, according to the annotation guidelines for creating the gold standard, medications occurring under certain sections (e.g., family history and allergic reaction) should be excluded from the system output. Furthermore, knowing the types of sections could be useful for field detection and field linking; for example, entries in the 'DISCHARGE MEDICATIONS' section are more likely to appear in a list of medications than in narrative text.

The set of sections and the exact spelling of section headings vary across discharge summaries. The section segmenter uses regular expressions (e.g., '^\s*([A-Z\s]+):' -- a line starting with a sequence of capitalized words followed by a colon) to collect potential section headings from the training data, and the headings whose frequencies are higher than a threshold are used to identify section boundaries in the discharge summaries.

### 3.2 Field detection

This step consists of three modules: the first module, *find_name*, finds medication names in a discharge summary; the second module, *context_type*, processes each medication name identified by *find_name* and determines whether the medication appears in narrative text or in a list of medications; the third module, *find_others*, detects the five non-name field types.

For *find_name* and *find_others*, we follow the common practice of treating named-entity (NE) detection as a sequence labeling task with the BIO tagging scheme; that is, each token in the input is tagged with B-x (beginning an NE of type x), I-x (inside an NE of type x) and O (outside any NE).

### 3.2.1 The *find_name* module

As this module identifies medication names only, the tagset under the BIO scheme has three tags: B-m for beginning of a name, I-m for inside a name, and O for outside. Various features are used for this module, which we group into four types:

- (F1) includes word n-gram features (n=1,2,3). For instance, the bigram $w_{i-1}$ $w_i$ looks at the current word and the previous word.
- (F2) contains features that check properties of the current word and its neighbors (e.g., the POS tag, the affixes and the length of a word, the type of section that a word appears in, whether a word is capitalized, whether a word is a number, etc.)
- (F3) checks the BIO tags of previous words
- (F4) contains features that check whether n-grams formed by neighboring words appear as part of medication names in given medication name lists. The name lists can come from labeled training data or additional resources such as UMLS.

### 3.2.2 The *context_type* module

This module is a binary classifier which determines whether a medication name occurs in a list or narrative context. Features used by this module include the section name as identified by the preprocessing step, the number of commas and words on the current line, the position of the medication name on the current line, and the current and near-by words.

### 3.2.3 The *find_others* module

This module complements the *find_name* module and uses eleven BIO tags to identify five non-name fields. The feature set used in this module is very similar to the one used in *find_name* except that some features in (F2) and (F4) are modified to suit the needs of the non-name fields. For instance, a feature will check whether a word fits a common pattern for dosage. In addition, some features in (F2) look at the output of previous modules: e.g., the location of nearby medication names as this information can be provided by the *find_name* module at test time.

## 3.3 Field linking

Once medication names and other fields have been found, the final step is to form entries by associating each medication name with its related fields. Our current implementation uses simple heuristics. First, we go over each non-name field and link it with the closest *preceding* medication name unless the distance between the non-name field and its closest *following* medication name is much shorter. Second, we assemble the (name, non-name) pairs to form medication entries with a few rules.

More information about the modules discussed in this section and features used by the modules is available in (Halgrim, 2009).

## 4 Experimental results

In this section, we report the performance of our system on the development set (Section 4.1-4.3) and the test set (Section 4.4). The data sets are described in Table 2. For all the experiments in this section, unless specified otherwise, we report exact horizontal precision/recall/F-score, the primary metrics for the i2b2 challenge.

For the three modules in the field detection step, we use the Maximum Entropy (MaxEnt) learner in the Mallet package (McCallum, 2002) because, in general, MaxEnt produces good results without much parameter tuning and the training time for MaxEnt is much faster than more sophisticated algorithms such as CRF (Lafferty et al., 2001).

To determine whether the difference between two systems' performances is statistically significant, we use approximate randomization tests (Noreen, 1989) as follows. Given two systems that we would like to compare, we first calculate the difference between exact horizontal F-scores. Then two pseudo-system outputs are generated by randomly swapping (at 0.5 probability) the two system outputs for each discharge summary. If the difference between F-scores of these pseudo-outputs is no less than the original F-score difference, a counter, *cnt,* is increased by one. This process was repeated n=10,000 times, and the p-value of the significance is equal to *(cnt+1)/(n+1)*.

If the p-value is smaller than a predefined threshold (e.g., 0.05), we conclude that the difference between the two systems is statistically significant.

### 4.1 Performance of the whole system

#### 4.1.1 Effect of feature sets

To test the effect of feature sets on system performance, we trained *find_name* and *find_others* with different feature sets and tested the whole system on the development set. For (F4), we used two medication name lists. The first list consists of medication names gathered from the training data. The second list includes drug names from the FDA database (www.accessdata.fda.gov/scripts/cder/ndc/). We use the second list to test whether adding features that check the information in an additional resource could improve the system performance.

The results are in Table 4. For the last two rows, F1-F4a uses the first medication name list, and F1-F4b uses both lists. The F-score difference between all adjacent rows is statistically significant at $p \leq 0.01$, except for the pair F1-F3 vs. F1-F4a. It is not surprising that using the first medication name list on top of F1-F3 does not improve the performance, as the same kind of information has already been captured by F1 features. The improvement of F1-F4b over F1-F4a shows that the system can easily incorporate additional resources and achieve a statistically significant (at $p \leq 0.01$) gain.

| Features | Precision | Recall | F-score |
|---|---|---|---|
| F1 | 72.5 | 60.3 | 65.8 |
| F1-F2 | 82.5 | 78.2 | 80.3 |
| F1-F3 | 88.4 | 77.9 | 82.8 |
| F1-F4a | 87.4 | 77.9 | 82.4 |
| F1-F4b | 88.1 | 79.4 | 83.5 |

Table 4: System performance on the development set with different feature sets

#### 4.1.2 Effect of training data size

Figure 1 shows the system performance on the development set when different portions of the training set are used for training. The curve with "+" signs represents the results for F1-F4b, and the curve with circles represents the results for F1-F4a. The figure illustrates that, as the training data size increases, the F-score with both feature sets improves. In addition, the additional resource is most helpful when the training data size is small, as indicated by the decreasing gap between the two sets of F-scores when the size of training data increases.

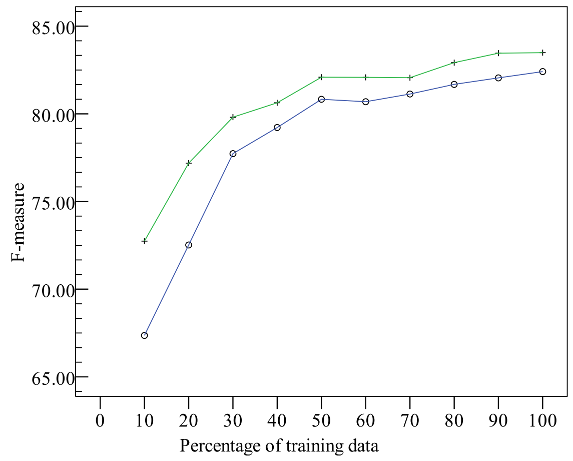

Figure 1: System performance on the development set with different training data sizes (Legend: ○ represents F-scores with features in F1-F4a; + represents F-scores with features in F1-F4b)

#### 4.1.3 Pipeline vs. *find_all*

The current field detection step is a pipeline approach with three modules: *find_name*, *context_type*, and *find_others*. Having three separate modules allows each module to choose the features that are most appropriate for it. In addition, later modules can use features that check the output of the previous modules. A potential downside of the pipeline system is that the errors in the early module would propagate to later modules. An alternative is to use a single module to detect all six field types together.

Figure 2 shows the result of *find_all* in comparison to the result for the three-module pipeline. Both use the F1-F4b feature sets, except that *find_others* uses some features that check the output of previous modules which are not available to *find_all*.

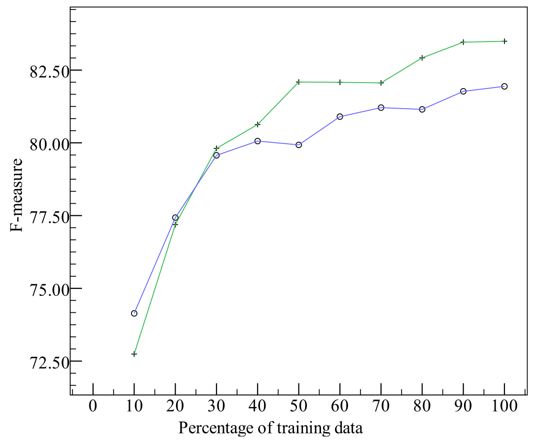

Figure 2: Pipeline vs. *find_all* for field detection (Legend: ○ represents F-scores with *find_all*; + represents F-scores with the three-module pipeline)

Interestingly, when 10% of the training set is used for training, *find_all* has a higher F-score than the pipeline approach, although the difference is not statistically significant at $p \leq 0.05$. As more data is used for training, the pipeline outperforms *find_all*, and when at least 50% of the training data is used, the difference between the two is statistically significant at $p \leq 0.05$. One possible explanation for this phenomenon is that as more training data becomes available, the early modules in the pipeline make fewer errors; as a result, the disadvantage of the pipeline approach caused by error propagation is outweighed by the advantage that the later modules in the pipeline can use features that check the output of the earlier modules.

## 4.2 Performance of the field detection step

Table 5 shows the *exact* precision/recall/F-score on identifying the six field types, using all the training data, F1-F4b features, and the pipeline approach for field detection. A span in the system output *exactly matches* a span in the gold standard if the two spans are identical and have the same field type. Among the six fields, the results for duration and reason are the lowest. That is because duration and reason are longer phrases than the other four field types and there are fewer strong, reliable cues to signal their presence.

When making the narrative/list distinction, the accuracy of our *context_type* module is 95.4%. In contrast, the accuracy of the baseline (which treats each medication name as in a list context) is only 55.6%.

| | Precision | Recall | F-score |
|---|---|---|---|
| Name | 91.2 | 88.5 | 89.9 |
| Dosage | 96.6 | 90.8 | 93.6 |
| Frequency | 93.9 | 89.0 | 91.8 |
| Mode | 95.7 | 90.3 | 92.9 |
| Duration | 73.8 | 43.2 | 54.5 |
| Reason | 72.2 | 31.0 | 43.3 |
| All fields | 92.6 | 84.5 | 88.4 |

Table 5: The performance (exact precision/recall/F-score) of field detection on the development set.

## 4.3 Performance of the field linking step

In order to evaluate the field linking step, we generated a list of (name, non-name) pairs from the gold standard, where the name and non-name fields appear in the same entry in the gold standard. We then compared these pairs with the ones produced by the field linking step and calculated precision/recall/F-score. Table 6 shows the result of two experiments: in the cheating experiment, the input to the field linking step is the fields from the gold standard; in the non-cheating experiment, the input is the output of the field detection step. These experiments show that, while the heuristic rules used in this step work reasonably well when the input is accurate, the performance deteriorates considerably when the input is noisy, an issue we plan to address in future work.

| | Precision | Recall | F-score |
|---|---|---|---|
| Non-cheating | 87.4 | 75.1 | 80.8 |
| Cheating | 96.2 | 94.5 | 95.3 |

Table 6: The performance of the field linking step on the development set (cheating: assuming perfect field input; non-cheating: using the output of the field detection step)

## 4.4 Results on the test data

Table 7 shows the system performance on the i2b2 official test data. The system was trained on the union of the training and development data. Compared with the top five i2b2 systems (see Table 3), our system was second only to the best i2b2 system, which used more resources and more sophisticated rules for field linking (Patrick and Li, 2009).

| | Precision | Recall | F-score |
|---|---|---|---|
| **Horizontal** | **88.6** | **80.2** | **84.1** |
| Name | 92.6 | 87.1 | 89.8 |
| Dosage | 96.3 | 90.2 | 93.1 |
| Frequency | 95.6 | 90.8 | 93.2 |
| Mode | 96.7 | 90.2 | 93.3 |
| Duration | 70.6 | 40.5 | 51.5 |
| Reason | 73.4 | 34.7 | 47.1 |
| All fields | 91.6 | 82.7 | 86.9 |

Table 7: System performance on the test set when trained on the union of the training and the development sets with F1-F4b features.

## 5 Conclusion

We present a hybrid system for medication extraction. The system is built around a pipeline of cascading statistical classifiers for field detection. It achieves good performance that is comparable to the top systems in the i2b2 challenge, and incorporating additional resources as features further improves the performance. In the future, we plan to replace the current rule-based field linking module with a statistical module to improve accuracy.

## Acknowledgments

This work was supported in part by US DOD grant N00244-091-0081 and NIH Grants 1K99LM010227-0110, U54LM008748, and T15LM007442-06. We would also like to thank three anonymous reviewers for helpful comments.

## References

Trond Grenager, Dan Klein, and Christopher Manning. 2005. Unsupervised learning of field segmentation models for information extraction. In Proc. of ACL-2005.

Scott Halgrim. 2009. A Pipeline Machine Learning Approach to Biomedical Information Extraction. Master Thesis. University of Washington.

J. Lafferty and A. McCallum and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. of the 18th International Conference on Machine Learning (ICML-2001).

Matthew A. Levin, Marina Krol, Ankur M. Doshi, and David L. Reich. 2007. Extraction and mapping of drug names from free text to a standardized nomenclature. *AMIA Symposium Proceedings,* pp 438-442.

S. Meystre, G. Savova, K. Kipper-Schuler, and J. Hurdle. 2008. Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. *IMIA Yearbook of Medical Informatics Methods Inf Med 2008*; 47 Suppl 1:128-44.

Andrew McCallum. 2002. Mallet: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu

Eric W. Noreen. 1989. Computer intensive methods for testing hypotheses: an introduction. John Wiley & Sons.

Jon Patrick and Min Li, 2009. A Cascade Approach to Extract Medication Event (i2b2 challenge 2009). Presentation at the Third i2b2 Workshop, November 2009, San Francisco, CA.

Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In Proc. of the National Conference on Artificial Intelligence (AAAI), pp 913-918.

Benjamin Rozenfeld and Ronen Feldman. 2008. Self-supervised relation extraction from the web. Knowledge and Information Systems, 17(1):17-33.

Özlem Uzuner, Imre Solti, and Eithon Cadag, 2010a. Extracting Medication Information from Clinical Text. Submitted to JAMIA.

Özlem Uzuner, Imre Solti, Fei Xia, and Eithon Cadag, 2010b. Community Annotation Experiment for Ground Truth Generation for the i2b2 Medication Challenge. Submitted to JAMIA.

Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. 2004. An integrated, conditional model of information extraction and coreference with application to citation matching. In Proc. of the 20th Conference on Uncertainty in AI (UAI-2004).