# A General Supervised Approach to Segmentation of Clinical Texts

Kavita Ganesan
3M Health Information Systems
575 West Murray Blvd
Salt Lake City, UT
Email: kganesan3@mmm.com

Michael Subotin
3M Health Information Systems
12215 Plum Orchard Drive
Silver Spring, MD 20904
Email: msubotin@mmm.com

***Abstract*—Segmentation of clinical texts is critical for all sorts of tasks such as medical coding for billing, auto drafting of discharge summaries, patient problem list generation and many such applications. While there have been previous studies on using supervised approaches to segmentation of clinical texts, these existing approaches were trained and tested on a fairly limited data set showing low adaptibility to new unseen documents. We propose a highly generalized supervised model for segmenting clinical texts, based on a set of line-wise predictions by a classifier with constraints imposing their coherence. Evaluation results on 5 independent test sets show that our approach can work on all sorts of note types and performs consistently across enterprises.**

## I. Introduction

With the federal mandate on all health records going electronic to be in effect soon, the growth of Electronic Medical Record (EMR) data is going to be explosive. This will trigger the development of all sorts of health informatics applications for improved clinical outcomes and reduction in health care costs which was previously not possible as clinical notes were still on paper or stored using word processing applications.

Depending on the type of visit or treatment, clinical notes can consist of all sorts of information including patient demographics, medical history, family history, past surgeries, current medications, allergies and many such details. One of the core pieces in making sense of clinical notes and leveraging these notes for all sorts of data analysis is the ability to use parts of the clinical notes that are of interest. For example, for analysis of the types of allergies that a population of patients may have, we may only need to use the 'Allergy' section of a clinical note ignoring all other sections. Another example is auto-coding for billing purposes. When generating ICD billing codes for diagnosis, we only need to consider sections that may yield diagnosis codes. For example, 'chief complaint', 'discharge diagnosis' and 'hospital course' could be important candidate sections for diagnosis coding. In contrast, sections that are procedure related (e.g. 'Technique' or 'Procedures performed') or lab related sections (e.g. 'Laboratory data') may be much less important and can be ignored for this specific task. Using all sections in a clinical note could be potentially harmful as it can yield in false positive codes.

In order to be able to utilize the sections within clinical notes efficiently, we need a highly robust method to sectionize the notes into logical parts. While previous works [1][2] have explored the task of automatic segmentation of clinical texts, none of these approaches have been shown to work on a varied set of note types that cross different organizations. In fact, Tepper et al [2] showed that their segmentation model had high accuracy on the same data set but significantly lower accuracy on a separate data set. There could be several reasons for this including the types of documents used for training along with the features, preventing the model from generalizing sufficiently well.

Our goal in this paper is to be able to build a highly general segmentation model for clinical texts that is capable of identifying the *header*, *footer*, and all of the *top-level sections* of a clinical note (e.g. Allergies, Chief Complaint, Review of Systems, etc). With such a model, there will be no need to build a segmentation tool for individual hospitals or note types which would allow for application building over individual sections of interest. We propose an approach that uses an $\ell_1$-Regularized multi-class Logistic Regression model to classify each line according to five roles (i.e. start of a header, continuation of header, start of a section, continuation of section, footer). Since not all the sequences of these roles may make sense, we use the Viterbi algorithm [3] to find the highest-scoring well-formed sequence of line labels. Evaluation results across 5 test sets that cross different enterprises and note types show that we are able to achieve an average line-wise accuracy of 93.32% and a $P_k$ measure of 7.93%.

## II. Related Work

While much work has been done in segmentation of texts in general [4][5][6], work related to segmentation of clinical texts has been very limited. Related approaches have primarily focused on mapping sections to standard section types [7][8][9]. For example 'PMH', 'Previous medical history' and 'Past Medical History' all would map to a standard type such as *Past_Medical_History*. This problem is generally referred to as section classification. It is assumed that the sections themselves can be obtained using simple heuristics and regular expression. This can be true if the heuristics are applied to clinical notes within one organization and for specific note types. However, such heuristics may break when notes from a different organization or unseen note types may need to be

segmented. In Section III, we show some of the difficulties in segmenting clinical notes.

The only known works that have formally studied segmentation of clinical texts are the works of Tepper et al [2] and Apostolova et al [1]. While both these approaches use a machine learning based approach to detect section boundaries, these approaches differ in several ways compared to our work. First, these approaches were developed and tested on a very limited set of note types. Apostolova et al [1] for example, developed an SVM classifier trained and tested on radiology notes. Radiology notes are outpatient notes and have a fairly consistent format and a very concise structure compared to other clinical notes such as discharge summaries, history and physical and progress notes. Our model on the other hand is trained and tested on a wide range of note types including inpatient and outpatient notes. We also show a higher average accuracy than shown in Apostolova et al [1]. Tepper et al [2] trained their model using discharge summaries and radiology reports and report low adaptability when their model is applied to documents from a different data set than the training set. Our approach is robust to independent test sets from very different enterprises and can work across note types as would be shown in Section VI. Also, the features that we used to train our model are quite different from that proposed in Apostolova et al [1] and Tepper et al [2].

## III. Challenges in Segmenting Clinical Notes

Clinical notes generally follow the SOAP (Subjective, Objective, Assessment and Plan) style of writing where a section is typically represented by a *section header* that corresponds to one of the four categories within the SOAP structure. For example, we can have a section titled *Subjective* that directly maps to the Subjective category in the SOAP structure. You can also have other titles such as *Chief Complaint* and *History of Present Illness* that map to the Subjective component. Each section header often consists of a cue that separates the content of the section from the title. This is sometimes in the form of a colon (e.g. Allergies:) or sometimes the section header alone is capitalized (e.g. ALLERGIES). There are also many cases where the content of the section is placed right below the section header (with or without capitalization of the header text). Thus, oftentimes we have some formatting cues that can be leveraged to segment out these documents. This is unlike other types of unstructured text such as emails and news articles where formatting cues are almost never present. Figure 1 shows an example of an admit note with some of the issues typically present within clinical documents.

One would argue, that simple regular expressions and exact header matches (from a header repository) may suffice for this special segmentation task as we have clues like 'colons' that can be used and there are only so many header variations in clinical notes. Based on Figure 1, we point out some difficulties where regular expressions or exact header matches would easily break under various circumstances.

**Inconsistent header structure and naming.** First, the use of capitalization and colons with the header text can be very inconsistent within the same document let alone across documents, physicians and enterprises. For example, in Figure 1 we have the headers *Chief Complain* (mixed case) + colon followed by *PMH and Social Hx* (mixed case) + colon followed by *Allergies* (title case) + colon + negative sign followed by *Medications* (title case). If we rely only on the presence of colon to find potential headers, this would capture 3 of the 4 main headers leaving out *Medications* and would also capture patient demographics such as *Date of admission*, *D.O.B*, etc. So its clear that regular expressions could easily fail even in such simple cases and can significantly overproduce sections. Another important point to note is that, in some cases, two headers that usually appear individually may be grouped together. For example, *PMH* and *Social Hx* usually appear as separate sections. However, in Figure 1, they are grouped together. Thus, looking for exact header matches would not work for all cases.

```
ADMIT_NOTE *
Patient Name: xxx   D.O.B: xxx
Date of admission: xxx

Chief Complain: Difficulty breathing

PMH and Social Hx:
Patient has history of asthma. Patient
is a non-smoker.

Allergies :- None
Medications
-Albuterol
```

Fig. 1. Sample admit note with some of the challenges in segmenting such a note.

**Header mis-spellings.** Next, the headers in clinical notes are prone to mispellings (e.g. 'Chief Complain') which is usually unavoidable due to human error. Exact header matches will thus easily break in this case.

**Inconsistent content formatting.** Another very common problem is that there can be sudden non-logical line breaks in clinical texts. This could be due to issues related to Optical Character Recognition (OCR), where some of the older notes which were on paper are brought into the electronic world. Physicians may also choose to place line breaks as desired since they are essentially working with free-text. Even the spacing used throughout a document can be very inconsistent. For example, we may have multiple empty line spaces between two sections in one case and a single empty line space in another case. We can also have multiple empty line spaces between lines within the same section.

To be robust to many of the issues presented in this section, we propose to use a supervised segmentation model, with a novel feature set and a generalizable training set. In Section IV we describe our approach in detail.

```
[BH] ADMIT_NOTE *
[CH] Patient Name: xxx   D.O.B: xxx
[CH] Date of admission: xxx
[CH]
[BS] Chief Complain: Difficulty breathing
[CS]
[BS] PMH and Social Hx:
[CS] Patient has history of asthma. Patient
[CS] is a non-smoker.
[CS]
[BS] Allergies :- None
[BS] Medications
[CS] -Albuterol
```

Fig. 2. Sample admit note that has been tagged with section labels. BH=BeginHeader; BS=BeginSection; CH=ContHeader; CS=ContSection.

## IV. Methods

### A. High-level Overview

In building our supervised model, we train a Logistic Regression Model with $\ell_1$ regularization using training data selected as described in Section V-C along with features as described in Tables I and II. We focus on detecting *top-level sections*, *header* and *footer* within clinical notes. A *header* in a clinical note is usually at the start of a note often containing the title of the note and patient demographics such as name, age, date of exam, accession number, etc. *Footer* on the other hand is at the end of the note and usually contains information about the medical staff. While clinical documents typically start with a header and end with a footer, there are some documents that contain very minimal header or footer information or non at all. *Top-level sections* are sections within the body of a clinical note which are not sub-sections of any given section. To find sections within a clinical note, we scan the document on a line-by-line basis and label each line to contain one of the following 5 labels:

1) BeginHeader - start of document header
2) ContHeader - continuation of document header
3) BeginSection - start of a top-level section
4) ContSection - continuation of a top-level section
5) Footer - footer part of the document

We do not distinguish between the start and continuation of document footers, because they do not generally start with a distinctive element. After a probability distribution over the set of labels is obtained for each line in a clinical document, we determine the highest-scoring well-formed label sequence using the Viterbi algorithm [3], which serves to eliminate incompatible label pairs, as described in more detail in Section IV-C. While this may seem like a less sophisticated analogue of CRFs, we find that our approach not only scales better at training, but also achieves higher accuracy than a standard linear-chain CRF on our task.

Figure 2 shows the example admit note shown earlier labeled with the potential 5 labels. With this labeling strategy, we see that it would be easy to obtain the sections of interest including the header and the footer. In the next few sections, we will discuss the specific tools and components used in building our segmentation model.

### B. $\ell_1$-Regularized Logistic Regression

Logistic regression is widely used in machine learning for all sorts of classification problems and is especially popular for text classification. It is well-known that regularization is required to avoid over-fitting [10], especially when there is only a small number of training examples, or when there are a large number of parameters to be learned.mWe specifically used the Logistic Regression implementation from the Liblinear library [11].

### C. Constrained model combination

The simplest way to combine the line-wise classifier predictions at run time is to pick the most probable label for each line. However, this naïve approach could easily produce a sequence of labels that does not correspond to a coherent region structure for the document[1]. For example, a *ContSection* label cannot follow any labels except *BeginSection* and *ContSection*. To ensure that the resulting label sequence is well formed, we impose the following constraints on consecutive pairs of labels:

1) The first line of the document must be *BeginHeader* or *BeginSection*;
2) *BeginHeader* cannot be preceded by *BeginHeader* or *ContHeader*;
3) *ContHeader* must be preceded by *BeginHeader* or *ContHeader*;
4) *ContSection* must be preceded by *BeginSection* or *ContSection*;
5) *Footer* cannot be preceded by *BeginHeader* or *ContHeader*.

Constraint 1 ensures that the start of a document coincides with the start of a header or section. Constraint 2 ensures that there are no spurious subdivisions of headers. Constraints 3 and 4 ensure that each header and section has a beginning label. Finally, constraint 5 ensures that there is at least one section between a header and a footer. Note that these constraints do not prevent a document from having multiple headers and footers. This restriction was eliminated because we observed that some clinical notes are concatenations of multiple documents, each with its own header and footer.

The general idea of finding an optimal combination of classifier predictions subject to a set of declarative constraints has been applied in many settings, and various methods for solving this class of problems have been proposed [12]. In this particular case, it can be solved by an efficient dynamic programming algorithm. To describe it, we will introduce some notation.

[1]The same challenge was faced by Tepper et al [2], who note that they used beam search to find a good label sequence without giving further detail.

Consider a document containing $N$ lines and let the variables $x_1 \dots x_N$ represent their sequence. Let $G(x_i)$ denote the structural role label assigned by a classifier to line $x_i$ by and let $g(G(x_i))$ be its classification score. Let us define a scoring function $S(x_1 \dots x_N)$ for a given assignment of labels over an entire document, which incorporates both the classification scores and the constraints as follows:

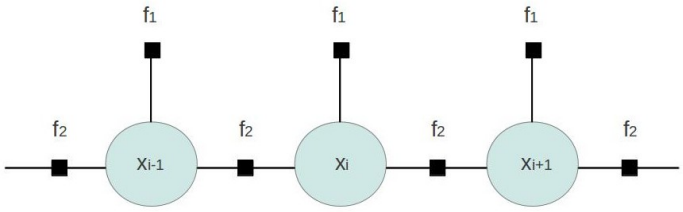
$$S(x_1 \dots x_N) = \prod_{i=1}^{N} g(G(x_i)) \times \prod_{i=1}^{N-1} \prod_{j=1}^{5} c_j(x_i, x_{i+1}) \quad (1)$$

where the terms $c_j(x_i, x_{i+1})$ represent the five constraints. For example, the first constraint term would be defined as

$$c_1(x_i, x_{i+1}) = \begin{cases} 0 & \text{if } G(x_i) \text{ and } G(x_{i+1}) \\ & \text{violate constraint 1} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

Thus, the sequence with the highest score of $S(x_1 \dots x_N)$ automatically satisfies all the constraints, since any sequence violating at least one constraint would have a score of zero. We can represent the total document score by a factor graph [13] shown in Figure 3, containing the two factor types $f_1(x_i) = g(G(x_i))$ and $f_2(x_i, x_{i+1}) = \prod_{j=1}^{5} c_j(x_i, x_{i+1})$. Since this is a linear-chain factor graph, the highest-scoring sequence $S(x_1 \dots x_N)$ can be computed by the Viterbi algorithm.

Fig. 3. Factor graph of the model.




### D. Identifying Candidate Headers using a Header Inventory

One of our key source of features is based on an inventory of known headers. This is essentially a collection of header names compiled over notes that we have worked with where some of these headers actually have mappings to *canonical types*. For example, 'PMH' and 'Previous medical history' would map to a canonical type called *Past_Medical_History*. We have 471 such canonical types and 22,611 headers in total in our header inventory. While this specific resource is proprietary, a similar resource can be created by collecting headers from a sample of clinical notes and then the canonical types can be manually or semi-automatically assigned.

We apply a rule-based module that uses exact substring matching and regular expressions to find potential headers and variants of these headers using the header inventory as a basis on any given clinical document. This is a high recall module as it not only finds potential headers in top-level sections but also sub-sections, headers and footers. When we have a potential header match, we then obtain the canonical type (if present) and use both the name of the matched header and the corresponding canonical type to build part of our features as we describe in detail in Section IV-E1.

### E. Model Features

We use a fairly well studied set of features in this paper. Tables I and II provide a high-level summary of all the features that we experimented with.

*1) KnownHeader Features:* One of the core features that we use in our model are the KnownHeader features. KnownHeader features are features derived from a header inventory described in Section IV-D. Below we describe our list of KnownHeader features in detail.

**HeaderBasic:** This feature is a combination of 3 aspects.
(1) Normalization, where the header text is lowercased with special characters, determiners and numbers removed;
(2) Canonical type of header (only if present);
(3) A marker to indicate that there was a match in the header inventory.

**HeaderUnigram:** Unigram tokens of normalized header text.

**HeaderCharNGram3:** Character n-grams of length up to 3 of normalized header text.

**CapsColon:** This is a combination of 3 aspects.
(1) Capitalization info of header text (all caps, mixed case, capitalized, lower case);
(2) Colon: present or absent;
(3) Does this capitalization and colon type appear in the majority or minority of known headers in the note?

**HeaderLen:** Information on length of the header text. Is the header text less than 4, 5, 6 characters long? Is the header text more than 25, 50, 100 characters long?

Table II outlines the KnownHeader feature subsets that we use in our experiments.

*2) Other Features:* We also use other important features in training our segmentation model and as we will show later, these features are indeed effective. We summarize all features used in our experiments in Tables I and II.

**LineUnigram** - Unigram tokens of all text on a line. Each line is lowercased and non-letter characters removed.

**PosInDoc** - Relative position of a line in the document. This is a categorical attribute which maps to *top*, *middle* or *bottom*. We first compute the position in a document as: $pos = \frac{lineNo}{totalLines}$. If $pos \leq 0.3$ it is considered *top*; if $pos \geq 0.6$ it maps to *bottom*; otherwise it is considered *middle*.

**LineLenChange** - Length change from one line to another. This is a categorical attribute which maps to *same*, *shorter* or *longer*. If the length difference between the current line and and the previous line is less than 50 characters long (positive or negative), then this maps to *same*. If the absolute length difference is larger than 50 characters and the current line is longer than the previous, then this maps to *longer* otherwise it maps to *shorter*.

TABLE II
KNOWNHEADER FEATURE GROUPS

| KnownHeader Name | Known Header Features |
|---|---|
| KnownHeader1 | HeaderBasic |
| KnownHeader2 | HeaderBasic + HeaderUnigram |
| KnownHeader3 | HeaderBasic + HeaderUnigram + HeaderCharNGram3 |
| KnownHeader4 | HeaderBasic + HeaderUnigram + HeaderCharNGram3 + CapsColon |
| KnownHeader5 | HeaderBasic + HeaderUnigram + HeaderCharNGram3 + CapsColon + HeaderLen |

TABLE III
STATISTICS OF DATASET USED FOR TRAINING AND TESTING

| Dataset Name | Enterprise(s) | Description | # Notes | # Train Notes | # Test Notes | Avg # Sections | # Note Types |
|---|---|---|---|---|---|---|---|
| Rad1MixedHosps | Mixed | Radiology and Surgery Notes (outpatient) | 9000 | - | 9000 | 4 | 2 |
| Rad2HospA | Single | Radiology Notes (outpatient) | 1902 | - | 1902 | 6 | 1 |
| Rad3MixedHosps | Mixed | Radiology Notes (outpatient) | 100 | 0-100 | - | 4 | 2 |
| Inp1HospB | Single | Discharge summary, History and Physical | 300 | - | 300 | 9 | 2 |
| Inp2HospC | Single | A whole range of inpatient notes with notes that are generally much longer, sampled from a larger set of notes | 1000 | - | 1000 | 16 | 10 |
| Inp3HospD | Single | A whole range of inpatient notes randomly sampled from a much larger set | 2000 | 100-1700 | 300 | 10 | 11 |
| **Total** | | | **14302** | **100-1800** | **12502** | - | - |

TABLE I
HIGH-LEVEL SUMMARY OF FEATURES USED

| **Known header related features** | |
|---|---|
| **Feature Name** | **Feature Description** |
| HeaderBasic | Normalized header text, canonical name, dictionary match marker |
| HeaderUnigram | Unigram of normalized header text |
| HeaderCharNGram3 | Char n-gram of length up to 3 of normalized header text. |
| CapsColon | Capitalization and colon info. |
| HeaderLen | Length of header text. |

| **Other features** | |
|---|---|
| **Feature Name** | **Feature Description** |
| LineUnigram | Normalized unigram of text on a line |
| PosInDoc | Relative position of line in document. |
| LineLenChange | Length change from one line to another. |

## V. EXPERIMENTS

### A. Dataset

Our data set consists of a wide range of clinical notes from both the inpatient and outpatient setting. Notes on the outpatient side differ in that they tend to be shorter and more concise than inpatient notes as they generally document specific procedures and one-time consultation. The average number of sections in our outpatient notes is 4.37 and on the inpatient site it is 11.67.

Table III provides a summary of the data set that we use for testing and training. Instead of relying on n-folded cross-validation accuracy, we use a small part of our data set for training and use the remaining majority of our data set for independent testing as shown in Table III. Without these independent test sets it would be extremely hard to understand the real performance of the segmentation model on unseen documents. We now briefly describe our dataset:

**Inp1HospB:** This dataset contains 300 notes (discharge summaries and history and physical) from a single enterprise. The notes were manually sectioned by subject matter experts and were primarily used as a test set other than to show how training on a limited set of note types can prevent model generalization.

**Inp2HospC & Inp3HospD:** Both these dataset are from the inpatient side and come from independent enterprises. Both were sampled from a much larger set of documents (4000 and 35,000 respectively) and contain a very varied set of notes. Inp3HospD contains 11 note types and Inp2HospC contains 10 different note types. These data set were semi automatically sectioned where sections were first detected using a high-recall regular expression sectioner. The sections for about 10-20 notes from each note type from each enterprise were then manually reviewed by subject matter experts and the sectioning rules were then tweaked and refined by these experts. Inp2HospC was used only for testing and Inp3HospD was used for both training and testing where we held out 300 documents for testing.

**Rad1MixedHosps:** This data set contains 9000 notes from the outpatient setting with a mixture of radiology and surgery

notes from mixed enterprises. These notes were sectioned by a rule based sectioner developed primarily for radiology and surgery notes. This rule-based regioner was optimized to achieve high accuracy for specific specialties (i.e. surgery and radiology) and enterprises but performed poorly in other contexts. This data set was primarily used for testing.

**Rad2HospA & Rad3MixedHosps:** Rad2HospA contains 1902 radiology notes from a single enterprise with multiple facilities and Rad3MixedHosps contains 100 radiology notes from mixed enterprises. As with Rad1MixedHosps, these notes were sectioned using the same rule-based sectioner developed primarily for radiology and surgery notes. Rad2HospA was used for testing and Rad3MixedHosps was used for training.

### *B. Evaluation Metric*

Since our classification problem is a multi-class classification problem, one of our evaluation metric is line-wise accuracy since we perform labelling on a line by line basis. We also use a standard measure used in evaluation of text segmentation tasks known as $P_k$ [4]. $P_k$ observes a scrolling window of text in the document and compares the number of segmentation boundaries in the reference and the system hypothesis. If they are different, a penalty is incurred. In other words, we are computing the probability that the numbers of hypothesis and reference boundaries are different in any given window throughout the text. Thus, lower values indicate better agreement with a gold standard segmentation.

To evaluate the performance of our model we use independent test sets made up of a diverse set of notes from various enterprises as described in Section V-A. We used 3-folded cross-validation for initial development tests. However, to select the best data set and features for training, we used our independent test sets.

### *C. Training Data Selection Strategy*

We believe that in order to build a highly generalized classifier that will work across enterprises (organizations) and note types, we need to train our classifier on a set of notes that are fairly diverse. This is because with such an approach, we would be providing the learning model clinical notes with variation in headers, different documentation styles and notes of different length. To select this diverse set of training data, we started off with 35,000 notes from Inp3HospD which contains 11 different note types ranging from discharge summaries to operative notes. From these 35,000 documents we randomly sampled 2000 documents out of which 300 were witheld for testing and 1700 were reserved for training. We used Inp3HospD because it has a wide variety of notes and its sectioning rules were carefully tweaked.

Using the 1700 documents from *Inp3HospD* and 100 docs from *Rad3MixedHosps* which contains radiology notes we selected the best combination of notes (including number of training notes) cross-validated using our independent test sets. Specifically, assumining $N$ is this size of the training set, we randomly draw a sample of $N$ notes, where 90% of notes come from Inp3HospD and 10% of notes come from Rad3MixedHosps. We obtain 10 random samples for each $N$. In our case, the training set size is $N \in [50, 1800]$ with an increment of 50 notes at each step. Thus, we would have 10 randomly sampled training sets of size $N = 50$, 10 training sets of size $N = 100$ and so on. Assuming $t_i$ refers to each individual training set, we train a classifier (using our best features) on each $t_i$ and we note the accuracy on all of our test sets. We then select the $t_i$ that yields the highest accuracy to be our final training set. We avoid relying on n-folded cross validation as it may not reflect true accuracy on unseen documents. Later in our results, we show that this training data selection strategy is indeed effective and we also show how training data size affects accuracy.

TABLE IV
ACCURACY DIFFERENCES TRAINING ON A LIMITED SET OF NOTE TYPES VS. A VARIED SAMPLE OF NOTE TYPES

| Training Set | Test Set | 3-Folded Cross-Eval Accuracy | Test Accuracy |
|---|---|---|---|
| **Inp1HospB** (300 docs) **-** limited | **Inp2HospC** (1000 docs) | 96.70% | 67.00% |
| **Inp3HospD** (300 docs) - varied | **Inp2HospC** (1000 docs) | 96.58% | 88.23% |

## VI. RESULTS

Our best model is trained on a combination of notes from *Inp3HospD* and *Rad3MixedHosps* with a total of 350 notes. The notes were selected using the data selection strategy outlined in Section V-C with the following features: *LineUnigram*, *PosInDoc*, *KnownHeader5* and *LineLenChange*. Unless otherwise mentioned, we report results using our best model.

**Performance differences using limited notes vs. a varied set of notes for training.** One of our hypotheses in this paper is that if we train our model on a limited set of note types as done by Tepper et al [2] and Apostolova et al [1], the segmentation model would not generalize enough to detect boundaries in a wide range of notes. To test this hypothesis, we trained our model using only *Inp1HospB* which contains discharge summaries and history and physical notes (note that this data was manually sectioned by experts) and we tested the model on *Inp2HospC*. Similarly, we trained a separate model using 300 notes from *Inp3HospD*, a data set from an independent enterprise containing a very varied set of notes and also tested on *Inp2HospC*. We use the *LineUnigram* and *KnownHeader5* features to train these models. We report performance differences in Table IV.

As can be seen, even though both models have a high 3-folded cross validation accuracy, the test accuracy achieved when the model is trained on a *limited set of note types* is significantlty lower than when trained on a *varied set of notes* (67% versus 88%). This confirms our hypothesis that a varied set of notes is required to help build a generalized segmentation model for clinical notes. Note that even though *Inp1HospB* was manually sectioned, it failed to provide enough variation to help the model learn the task of segmenting unseen formatting and headers in clinical notes.

**Performance stability across enterprises and note types.** In Table V we report performance on all 5 test sets which crosses

TABLE V
MODEL PERFORMANCE ACROSS ENTERPRISES COVERING INPATIENT AND OUTPATIENT DOCUMENTS

| Note Type | Accuracy | Pk |
|---|---|---|
| Rad1MixedHosps (9000) | 92.45% | 6.13% |
| Rad2HospA (1902) | 93.67% | 10.32% |
| Inp1HospB (300) | 92.58% | 5.18% |
| Inp2HospC (1000) | 93.39% | 18.31% |
| Inp3HospD (300 leave out) | 95.81% | 2.29% |
| **Average** | **93.32%** | **7.93%** |

TABLE VI
TEST ACCURACY ON INP2HOSPC ACROSS DIFFERENT NOTE TYPES USING OUR BEST MODEL

| Note Type | Accuracy |
|---|---|
| Progress Note | 87.80% |
| Operative Note | 92.20% |
| Consult Note | 94.60% |
| Physician Clinincals | 93.10% |
| Procedure Note | 83.60% |
| History and Physical | 95.70% |
| Short Stay Summary | 94.60% |
| Discharge Summary | 94.00% |
| Cardiac Cath | 85.40% |

multiple enterprises and contains a wide range of note types. Then, in Table VI we report performance by note type for our *Inp2HospC* test set. The results reported are based on the best feature set and best subset of training data.

First, based on Table V, observe that all of our test sets (both inpatient and outpatient) achieve an accuracy of above 92%. Also, note that the average line-wise accuracy is 93.32% and the $P_k$ measure is 7.93%. The $P_k$ measure essentially says that on average there is 7.93% probability that two segments in a clinical note are incorrectly identified as belonging to the same section. The accuracy on the other hand says that on average 93.32% of the lines would be correctly labelled. Since $P_k$ is consistently low and accuracy is consistently high across test sets, these values show that our segmentation model is stable and performs extremely well across enterprises.

Next, from Table VI, we can see that the performance is consistent across note types where most of the note types have accuracy above 85%. The lowest accuracy is on procedure notes and upon investigation we realized that some of the issues were due to low recall on labelling 'procedure' related sections as 'procedure' can also mean 'technique'. We ran an informal test by adding synonyms for procedure and techniques, where the term 'procedure' in a potential header can also mean 'technique' and vice versa. This acutally gave a boost in overall performance. We will further investigate the synonym aspect as part of our future work. From both Table V and VI it is thus clear that our model works well across enterprises and across note types which shows that our data selection strategy and features used for training the model is indeed effective.

**Is large amounts of data needed for training a general segmentation model?** When we have access to large amounts of data, it may seem like a supervised model trained on all of this data would yield the best performance. Figure 4 shows accuracy on our *Inp1HospB* test set with different sizes of training data. This model was trained on notes from *Inp3HospD* (which is one of our primary training sets) using our core features (LineUnigram+KnownHeader5). As can be seen, with just *100 notes* we already achieve an accuracy of 87%. The performance consistently improves as more notes are added and *peaks at 500 notes*. After this, the performance degrades and plateaus as more and more notes are introduced. This goes to show that using more data (or all available data) would not necessarily help build a robust segmentation model. What is more important is using the right combination of training notes along with effective features to help the model generalize well. In fact, our best model uses only 350 notes with the best set of features.

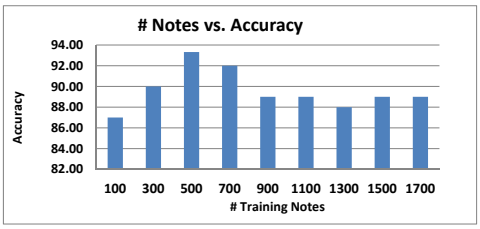



Fig. 4. Number of training notes versus accuracy on *Inp1HospB*. The model was trained on data from *Inp3HospD*.

**Feature analysis.** We now analyze the performance with the use of different features. Table VII shows the average accuracy on all of our test set as we incrementally add features. First, we can see that by just using a simple line-wise unigram, the model already yields an average accuracy of about 86% (FS1). This goes to show that our training set is very robust and our $\ell_1$-Regularized Logistic Regression model with label constraints (LR-LC) is very suitable for this task.

Further, note that when we add the PosInDoc feature (FS2), we get an additional 3.46% boost in accuracy. This shows that the relative position of a line within a clinical note is an important indicator for section boundaries. One reason for this could be because headers like 'History and Physical' can appear as a note header or a section header, and knowing their relative position in the document can help distinguish between these two possibilities.

Another very critical feature that we can observe from Table VII is the KnownHeader feature subset as defined in Table II. Notice that with the use of KnownHeader1-KnownHeader5 (feature set FS3-FS7), we see consistent improvement in accuracy over feature set FS2 up to 4.81%. Known headers may be providing cues on potential section headers and thus helping with the classifier's learning process. Although KnownHeader1 (FS3) is already quite effective, the best set of header features is one that uses all the KnownHeader features described in Section IV-E1 (FS7). Overall, we can see that using all of the proposed features gives us the best accuracy (FS8). Incrementally adding these features did not show any degradation in performance.

TABLE VII
AVERAGE TEST ACCURACY BY FEATURES

| Feature Set | Features | Avg Accuracy | Change |
|---|---|---|---|
| **FS1** | LineUnigram | 85.55% | |
| **FS2** | LineUnigram+PosInDoc | 88.62% | +3.46% |
| **FS3** | LineUnigram+PosInDoc+KnownHeader1 | 92.00% | +3.67% |
| **FS4** | LineUnigram+PosInDoc+KnownHeader2 | 92.05% | +3.72% |
| **FS5** | LineUnigram+PosInDoc+KnownHeader3 | 92.39% | +4.08% |
| **FS6** | LineUnigram+PosInDoc+KnownHeader4 | 92.67% | +4.37% |
| **FS7** | LineUnigram+PosInDoc+KnownHeader5 | 93.10% | +4.81% |
| **FS8** | LineUnigram+PosInDoc+KnownHeader5+LineLenChange | 93.32% | +0.24% |

TABLE VIII
ACCURACY COMPARISON ACROSS CLASSIFIERS.

| Test Set | LR-LC | NB-LC | SVM | CRF |
|---|---|---|---|---|
| Inp1HospB (300) | 92.58% | 82.60% | 75.80% | 86.20% |
| Inp2HospC (1000) | 93.39% | 88.30% | 88.00% | 71.50% |
| Inp3HospD (300 leave out) | 95.81% | 93.40% | 94.40% | 86.40% |
| **Average** | **93.32%** | **88.10%** | **86.07%** | **81.37%** |

**Comparison across different classifiers.** We now show how our $\ell_1$-Regularized Logistic Regression model with label constraints (LR-LC) performs in comparison with other classification models namely SVMs, CRFs and Naive Bayes. Note that with Naive Bayes we actually used the same label constraints as used in LR-LC and we refer to this as NB-LC. We report our comparison in Table VIII. Based on these results, we can see that our proposed approach, LR-LC, outperforms NB-LC, SVMs and CRFs. Even though CRFs are well known for sequence tagging problems, for this task the CRF model has the lowest average accuracy. The performance of the CRF model is especially low on Inp2HospC where the documents are on average longer than Inp1HospB and Inp3HospD (with an average of 16 sections vs. 9 and 10 sections respectively). Also note that the performance of classifiers that use our proposed label constraints (NB-LC and LR-LC) outperform SVMs and CRFs that do not have any imposed constraints. Thus, our constraint combination approach as explained in Section IV-C is indeed effective.

## VII. CONCLUSION

In this paper, we proposed a highly generalized supervised segmentation approach for clinical texts. We focused on discovering *top-level sections*, *header* and *footer* in arbitrary clinical texts. We proposed to solve this task using an $\ell_1$-Regularized Logistic Regression classifier with label constraints.

Our evaluation shows that our training set selection strategy and feature set are both robust for segmenting clinical texts across note types and organizations. We achieve an average accuracy of 93.32% and an average $P_k$ value of 7.93% across several test sets that span various enterprises both in the inpatient and outpatient setting (total of 12,502 test notes).

Our work can be further extended in several ways. First, the proposed header features can be further refined through the use of synonyms. For example, 'technique' can also mean 'procedure' and adding such synonyms can assist in the learning process. Next, while we have mainly encoutered documents that follow the standard SOAP style of writing, there are some organizations that have notes in tabular form. In the future, we would like to understand how our model can also cater to some of these non-traditional formatting.

## REFERENCES


[1] E. Apostolova, D. S. Channin, D. Demner-Fushman, J. Furst, S. Lytinen, and D. Raicu, "Automatic segmentation of clinical texts," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 5905–5908.

[2] M. Tepper, D. Capurro, F. Xia, L. Vanderwende, and M. Yetisgen-Yildiz, "Statistical section segmentation in free-text clinical records." in *LREC*, 2012, pp. 2001–2008.

[3] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.

[4] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Machine learning*, vol. 34, no. 1-3, pp. 177–210, 1999.

[5] F. Y. Choi, "Advances in domain independent linear text segmentation," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, 2000, pp. 26–33.

[6] H. Kozima, "Text segmentation based on similarity between words," in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1993, pp. 286–288.

[7] Y. Li, S. Lipsky Gorman, and N. Elhadad, "Section classification in clinical notes using supervised hidden markov model," in *Proceedings of the 1st ACM International Health Informatics Symposium*. ACM, 2010, pp. 744–750.

[8] J. C. Denny, R. A. Miller, K. B. Johnson, and A. Spickard III, "Development and evaluation of a clinical note section header terminology," in *AMIA Annual Symposium proceedings*, vol. 2008. American Medical Informatics Association, 2008, p. 156.

[9] J. C. Denny, A. Spickard III, K. B. Johnson, N. B. Peterson, J. F. Peterson, and R. A. Miller, "Evaluation of a method to identify and categorize section headers in clinical documents," *Journal of the American Medical Informatics Association*, vol. 16, no. 6, pp. 806–815, 2009.

[10] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient $l_1$ regularized logistic regression," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 401.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[12] M.-W. Chang, L. Ratinov, and D. Roth, "Structured learning with constrained conditional models," *Machine learning*, vol. 88, no. 3, pp. 399–431, 2012.

[13] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.