Contextualized Medication Event Extraction
National NLP Clinical Challenges (n2c2) 2022

Sabrina Sedovic
The University of Chicago
ssedovic@uchicago.edu

Jacob Lehr
The University of Chicago
jblehr@uchicago.edu

Michelle Orden
The University of Chicago
morden@uchicago.edu

Abstract

This project details our participation in the 2022 n2c2 challenge. The challenge's goal is to identify useful context categories from unstructured medical notes. Our solution involves several steps: (1) identify all mentions of medications within a given medical note, (2) for each medication mention, determine whether the medication mention involved disposition, and (3) for each disposition, classify the disposition across five different context dimensions. To address this challenge, we implement a series of token and sequence classification models.

1 Introduction

Patient information is increasingly captured in structured electronic health records. Official medication orders are often relied upon to track a patient's medication history. However, there are gaps in these official sources of data. Often, clinicians will provide informal verbal direction to patients to modify dosages or pause or restart medications, but do not necessarily order a new medication. Similarly, patients may report that they have modified their medication dosage or frequency on their own. All of this information is a crucial piece of a patient's medical history, but is often only recorded in unstructured clinical narratives, rather than official medication orders. When doctors then make decisions for new treatment plans, they may exclude this crucial source of information, or at the very least, find it more difficult to interpret.

In addition, doctors and other medical practitioners spend a large percentage of their time writing, reading, and interpreting their own (and other's) clinical notes in order to gain a complete understanding of their patient's medical history. This task, although important and necessary, can be cumbersome and onerous. The Natural Language Processing community would like to provide some relief to doctors and other medical practitioners by automating parts of this task. Computers and NLP models have the potential of processing clinical notes for medical practitioners to identify medication mentions as well as changes in medication prescriptions. In this way, NLP could aid medical practitioners by helping to fill gaps in official patient histories and allowing them to spend more time treating and interacting with patients rather than combing through text.

This report details our attempt at such a task. We use NLP methods to extract medication mentions, classify medication events (changes), and classify context information related to medication events. It is important to note that unlike other text, clinical text is a specific type of writing that is not seen in common places such as the web or in novels. Special models must be used in such analysis in order to properly capture the domain specific language used in clinical text.

2 Literature Review

As a first step in approaching this challenge, we reviewed several prior approaches to similar tasks. We also considered heavily the article written in concert with the task, which describes how the annotations were generated, and provides some initial modeling techniques.

In two articles, Mahajan et al. [2020], and Mahajan et al. [Forthcoming], the authors describe first the creation of a novel dataset, then attempt to classify medication events within clinical notes using various machine learning and natural language processing techniques. The authors present a new dataset, Contextualized Medication Event Dataset (CMED), which consists of 9,013 medication mentions annotated over 500 clinical notes. The clinical notes that served as the underlying data source for the authors were from 2014 i2b2/UTHealth Natural Language Processing shared task [Kumar et al., 2015]. The challenge here is that, although medical information is obtained through structured medication orders, many medication events are also documented in unstructured clinical notes. The authors attempt to create an organized schema of label definitions that captures both types and aspects of medication changes including temporal information (past, continuing, stop, start) and medication amount information (increase, decrease, no change).

The authors take a multi-step approach to classification. Their first task was to classify each medication mention within the notes into an event type:

- Disposition: change is being discussed
- NoDisposition: no change is being discussed
- Undetermined: more information is needed

Next, for each Disposition event, classify the event along five context dimensions:

- Action: What is the change discussed? (Start, Stop, Increase, Decrease, Other-Change, UniqueDose, Unknown)
- Temporality: When is this change intended to occur? (Past, Present, Future, Unknown)
- Certainty: How likely is this change to have occurred / will occur? (Certain, Hypothetical, Conditional, Unknown)
- Actor: Who initiated the change? (Physician, Patient, Unknown)

A summary of the overall tasks and each subtask can be seen in the annotation process outlined below in Figure 1.

In the 2020 paper, the authors use Support Vector Machines (SVM) with a linear kernel for their experiments. They use a number of features in their model, including n-grams, lexico-syntactic (part of speech tags and lemmatization), window-based, dependency-parse, note-section, and RxNorm features. They also implemented 5-fold cross validation on their training dataset. In the 2021 paper distributed along with the task, the best performing model was a ClinicalBERT model. We describe ClinicalBERT in more detail in Section 4, and provide a comparison of our model to Mahajan et al. results in Table 4.[Mahajan et al., Forthcoming]

3 Data Preparation

Our data set consisted of 400 clinical notes provided by n2c2. Of the 400 clinical notes, 350 notes were provided for training purposes and 50 notes were provided as a dev set. We did not have access to official test data, as this data will be released on May 2, 2022. To account for the lack of test data, we split the given 400 clinical notes into train, dev, and test sets. We held out 10 percent of both the 350 training notes and 50 dev notes. This resulted in a total of 315 training notes, 45 dev notes, and 40 test notes to use in our pipeline.

Clinical notes were provided in an unstructured *.txt* format. Associated with each *.txt* file was an accompanying *.ann* file, which contained the gold standard annotations/labels for the clinical note. In order to input such data into our models, we needed to convert the unstructured *.txt* files into a structured JSON representation. We did so using the following steps:

1. For each clinical note (*.txt* file), we used NLTK's PunktSentenceTokenizer to tokenize the note by sentence level. Additionally, we used this library to generate the start and end character positions for each sentence within the note.

2. For each associated *.ann* file, we used the provided *RecordTrack1* class to obtain a list of all medications within the file.

3. Depending on the specific task, we then prepared the data for input into the model by assigning the correct labels to each token, or list of tokens (in the case of sequence classification). The specific configuration of the input data will be discussed in more detail in Section 4.

4. Each sentence embedding, associated labels, and associated character span, was then appended to a JSON structure.

This process, performed on train, dev, and test files resulted in three JSON files for each task (and subtask) which could then be input into our models.

3.1 Summary of training data

Table 1 and Table 2 show a summary of the distribution of labels within our final training data.

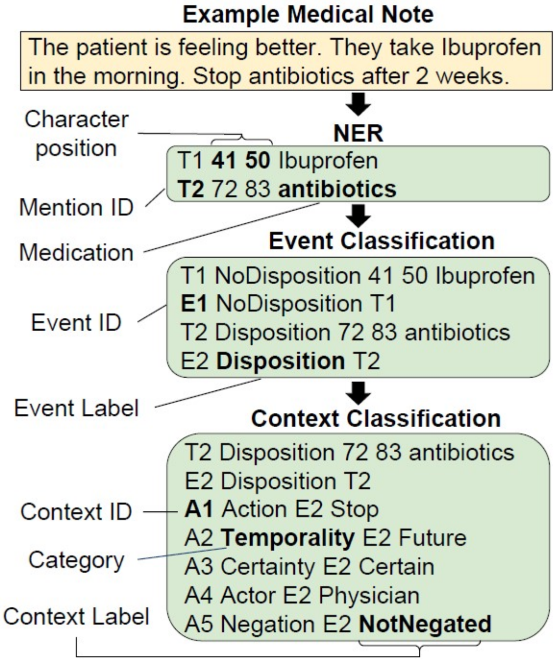

Figure 1: Annotation Process

As shown in Table 1, out of the 5,500 total medication mentions, less than 20 percent (1,070) are classified as a Disposition. These 1,070 medication mentions that include a disposition form the basis for all context classifications in Task 3. As shown in Table 2, for several context dimensions, we have highly imbalanced training data. For example, in the "Actor," "Negation," and "Certainty" context dimensions, over 90 percent of the labels are confined to one label. As will be discussed later, this imbalance may cause performance issues for some context dimensions.

4 Methods

Our system consisted of three main modeling steps, executed sequentially:

1. Classify each token into either medication mention or not;
2. For each medication, determine whether a change has been discussed;
3. For each medication with disposition, classify it along the five context dimensions.

The flow of data in our model can be seen in Figure 2. Note that this flow represents an "end-to-end" model design for the test set, where we use the predictions from the NER task as inputs to the Event task, and the predictions from the Event task as inputs to the Context task.

4.1 ClinicalBERT

Each model described below was initialized/trained on the ClinicalBERT model.

For each subtask, we used the ClinicalBERT model, a BERT model pre-trained specifically for clinical text. The ClinicalBERT model was pre-trained on all MIMIC III notes, a database containing electronic health records from ICU patients at the Beth Israel Hospital in Boston, MA, and initialized from BioBERT [Alsentzer et al., 2019]. We used the ClinicalBERT model because it performed best out of a range of models tried by Mahajan et al., and it is pre-trained on the exact same type of language that our model will see [Mahajan et al., Forthcoming].

4.2 Subtask 1: Medication Mention

The first step in classifying the context of medication mentions in medical notes is to identify where exactly medications are mentioned within each note. To address this task, we used a version of a token classification model called Named Entity Recognition (NER). NER models are often used to identify specifc named entities such as locations, people, or companies. We use a version of the "BIO" framework for labeling our data. For this application, we assign "B-MED" to the first token of each medication string, "I-MED" as the label for any additional tokens within the same medication, and "O" for all tokens that are not medications. We

Task/Subtask	Label	Count
NER	"Drug"	5,561
Event	NoDisposition	4,081
	Disposition	1,070
	Undetermined	411

Table 1: Gold standard label distribution in training set for NER and Event Tasks

Task/Subtask	Label	Count
Certainty	Certain	897
	Hypothetical	97
	Conditional	74
	Unknown	2
Negation	NotNegated	1,044
	Negated	26
Temporality	Past	546
	Present	395
	Future	103
	Unknown	26
Action	Start	414
	Stop	251
	UniqueDose	237
	Increase	102
	Decrease	36
	Unknown	29
Actor	Physician	968
	Patient	84
	Unknown	18

Table 2: Gold standard label distribution in training set for Context Task

predict all instances of "B-MED" or "I-MED".

One challenge in this classification task was in maintaining the correct number of words for a single medication mention. Since some medication mention span multiple words, and many medications spanned multiple tokens (e.g., "nitroglycerin" becomes four distinct tokens), we needed to ensure that our model only outputted a single label for each full mention.

We tried a number of different models before finalizing our approach:

1. Use individual tokens and labels as observations. This was not effective, as we encountered issues with the model only processing a single token at a time.

2. Group by document. We tried to pass each document (as a list of tokens) with corresponding labels as a single observation, but this was too large for the model to process.

3. Group by sentence. Finally, we ended up passing in a single sentence (as a list of words) with corresponding labels as a single observation, and this is ultimately the approach used in our model.

   More specifically, we first split each document into sentences, then tokenized each sentence using the WordPunctTokenizer from NLTK. Each observation is formed by a list of tokens. During the model training process, we then tokenize again using the tokenizer provided by the ClinicalBERT model, which breaks up words into multiple pieces.

We adapted the token classification framework from the Huggingface repo to work with the ClinicalBERT model to implement our token classification framework [Alsentzer et al., 2019].

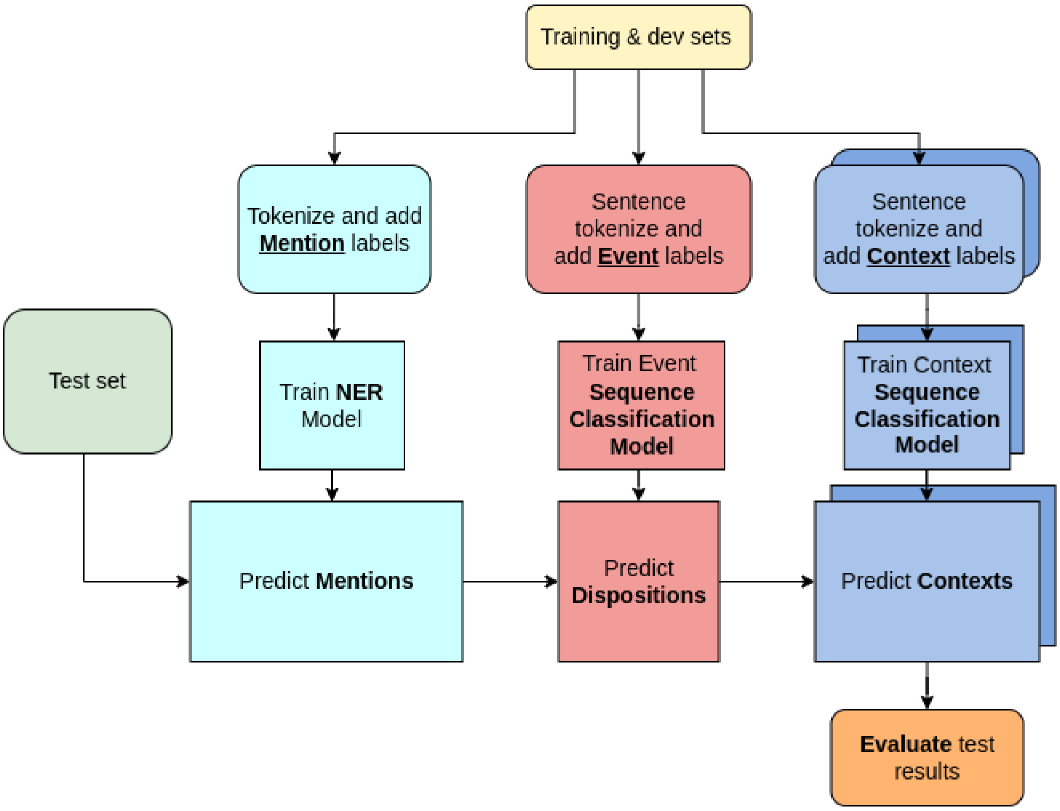
Training & dev sets
Tokenize and add Mention labels
Sentence tokenize and add Event labels
Sentence tokenize and add Context labels
Test set
Train NER Model
Train Event Sequence Classification Model
Train Context Sequence Classification Model
Predict Mentions
Predict Dispositions
Predict Contexts
Evaluate test results

Figure 2: Model Flow

4.2.1 NER Model specifications

For our final NER model, we trained using 3 epochs, a learning rate of $\epsilon = 0.00005$, and a batch size of 8.

4.3 Subtask 2: Event Classification

After a medication mention was found using the NER Model, it had to be determined whether a medication change was being discussed (Disposition) or not (No Disposition). Mentions that lacked clarity were labeled "Unknown". We initially attempted two different approaches: 1, to classify each token as either as Disposition, No Disposition, Unknown or not a medication; or 2, create a sequence surrounding the medication mentions found from the NER model, and then label each of these sequences as either Disposition, No Disposition, or Unknown depending on the medication classification. Approach 1 combined subtasks 1 and 2 into one task, while Approach 2 served as a continuation from subtask 1.

To perform Appraoch 1, we tokenized similarly to how we tokenized for subtask 1 (by sentence) and then labelled the tokens within the sentence as either "B-DIS"/"I-DIS" for disposition, "B-NOD"/"I-DOS" for no disposition, "B-UNK"/"I-UNK" for unknown, and "O" for any token that is not a medication. We then ran the NER Model on the data and labels to get a prediction for each medication.

To perform Approach 2, we first needed to create new datasets. Instead of tokenizing sentence by sentence and labelling each token, we created a sequence around each medication mention, consisting of the sentence containing the medication mention, along with the previous and subsequent sentences. The primary reason we included the other sentences was because the annotators were instructed to take into account that window when creating the annotations. We wanted to ensure our model was seeing the same information that the annotators were. We then labelled each sequence based on the medication mention's classification, "B-DIS" for disposition, "B-NOD" for no disposition, and "B-UNK" for unknown. Note that the same sequence might appear multiple times with multiple labels, as multiple medications could be mentioned in the same sentence. To classify a medication's disposition and context, we utilized Pytorch's Sequence Classification trained on ClinicalBERT. After labelling each sequence in the training and dev data, we inputted the sequences and labels into the Sequence Classification to find a model that best classified each sequence.

For the rest of our results, we used Approach 2. We chose to move forward with Approach 2 because whether a medication was classified as disposition or not widely depended on the context the medication was used in. The NER model, while performing similarly to the Sequence Classifier in terms of accuracy, does not necessarily rely on the context for the word to determine the disposition of a word. Instead, it recognizes whether a medication named is typically related to having disposition or not. Therefore, we chose to continue with the Sequence Classifier for the rest of our project in order to ensure the context of the medication mention was taken into account when determining the disposition. However, in further analysis, we would consider how other window sizes and adaptations to the NER model could be used to improve the accuracy of our event classification.

4.3.1 Sequence Classification specifications

For our final Event Sequence classification model, we used 5 epochs, a learning rate of $\epsilon = 0.00005$ and training/dev batch sizes of 16.

4.4 Subtask 3: Context Classification

A similar approach to Subtask 2 was taken for the Context classifications. For each of the five dimensions, we labeled each sequence of sentences surrounding a medication with disposition based on its sub-context. We then created five separate Sequence Classification models for each of the dimensions. It is important to note that the data available along the context dimensions was significantly lower than the ones for subtask 1 and subtask 2. Some labels, such as Negated in Negation, were hardly seen in the the annotations (as noted in Table 2). Thus, there was some difficulty in training the models, especially as small batch sizes had to be used. Negation, specifically, would predict all values as "Not Negated", including "Negated" values, leading to a 0 recall score and therefore a 0 F1 score on the dev set. It should be noted that Mahajan et al. [Forthcoming] did not report results on the Negation dimension because of this issue. We, however, still reported our results, since we wanted to demonstrate there is the possibility of creating a model for these situation, though it is difficult to perform better than the majority baseline.

After training the models, the test data was then used to evaluate the total performance of each of the models. That is, after running the Event task, we used the results from the task as input for the Context classification task, and evaluated the results against the gold standard test labels. We present the F1 scores in the Results section. We also ran the models separately from each other, inputting the original annotations to run through the context models, regardless of whether the medication mention was correctly labelled as a Disposition or not. These results are reported under the F1 Separate column in Table 4.

4.4.1 Sequence Classification specifications

The Context metrics are defined in Table 3. Note the learning rate for all models was .00005.

Context	Epochs	Batch Size
Action	3	8
Temporality	3	16
Certainty	5	8
Actor	5	16
Negation	3	16

Table 3: Metrics for Context Sequence Classifiers

5 Results

A summary of the results for overall tasks and each subtask can be found in Table 4. We provide results for two categories: (1) "end-to-end" and (2) "separate."

5.1 End-to-end system results

Our "end-to-end" results reflect the subtasks as outlined above in our model flow diagram, where each subtask depends on the predictions of the prior subtask. This is our best estimate of model performance in a "real-world" situation in which the goal is to make context classifications on raw text data (medical notes). At each stage, performance suffers because we no longer can predict on the entire dataset (the dataset now contains some false positives and negatives due to prior error). For example, after the NER model, we have some error because the model does not recognize 100 percent of the true medication mentions. When the subsequent Event model is run, it only makes predictions for medication mentions. Since some of these mentions are either missing or incorrect, the performance of the event classifier also suffers. This same trend carries into the subsequent context prediction task.

5.2 Separate model results

However, for the purposes of the competition and comparing our results to the Mahajan et al. [Forthcoming], we also present "separate" model results, where we evaluate each task in isolation. More specifically, the "separate" results reflect predictions made on test data that is complete as of the prior step. For example, for the event classification, our test data now includes all medication mentions within the gold standard annotation files. This allows us to evaluate the performance of each classifier in isolation. As expected, the performance is higher across every subtask and category when run in isolation.

5.3 Discussion

We can see that our overall (micro) F-1 scores for both the end-to-end and separate model pipelines for the Event classification task are very close to that of Mahajan et al. [Forthcoming]. Our F-1 scores for both the end-to-end and separate model pipelines for Action, Temporality, and Certainty classification tasks are lower than those of Mahajan et al. [Forthcoming]. Our F-1 scores for the end-to-end model for Actor classification task is also lower than that of Mahajan et al. [Forthcoming], however our score for the separate model for the Actor classification task is slightly higher.

It is useful to note that the paper's results are based off of a larger dataset consisting of 500 clinical notes, compared to our 400 notes. The paper also used a 75/5/20 percent train/dev/test split where we chose a 79/11/10 percent train/dev/test split. Having access to more training documents would also improve our results. There is some ambiguity as to whether or not the paper's system was set up in an "end-to-end" or "separate" style, but we assume they are separate due to better results in Task 3 for some context dimensions relative to Tasks 2.

5.4 Error Analysis

To identify some sources of errors in our model, we performed a basic error analysis. Shown below are several representative examples of incorrect context predictions, as well as a potential explanation for such errors.

Action Classification Error Example:

- Text: "11/93 w creat metformin D/C.A1c off it 3/94 8.5 and Glipizide inc to 10 mg, rec insulin but he deferred.Walks playing golf but stopped other exercise; discussed."
- Predicted value: Stop
- Actual value: Increase

As shown in this example, it is difficult to predict the action context, as there are often multiple actions within a span of 3 sentences. In this specific example, the key token here is "inc", as it signals an increase of a medication. However, within the context, we also see "stopped other exercise". In this case, it is likely that the model assigned more weight to the token "stopped" than the token "inc". This might make a case for a smaller context window in the future.

Temporality Classification Error Example:

- Text: "URIC ACID DONE TODAY.STARTED ON 0.6 COLCHICINE DAILY.IF URIC ACID HIGH WILL START ON LOW DOSE ALLOPURINOL IN VIEW OF PHOSPHO-TOPHI."
- Predicted value: Future
- Actual value: Past

This example shows that it can be difficult to separate the temporality of multiple medication mentions. In the text excerpt , there are multiple different time frames discussed (past, present, and future). It is common for multiple time periods to be discussed within the same context, and can result in the wrong temporal value being assigned to each medication mention.

Certainty Classification Error Example:

- Text: "Recheck today.I suspect we will be able to discontinue the glyburide.Urine albumin/creatinine ratio within normal limits six months ago."
- Predicted value: Certain
- Actual value: Hypothetical

This example shows a pervasive error throughout our analysis beyond just the Certainty dimension. In this example, there are multiple medication mentions: one which is correctly labeled as certain and one which is incorrectly labeled as certain. In this case, it is difficult to distinguish one medical mention from another, because the same contextual language is included in both classifications.

		F1	**F1**	**F1**
Task	**Subtask**	Ours	Ours	Mahajan, Liang, and Tsou, 2021
		End-to-end	**Separate**	**Separate**
NER	Medication Mention	0.97	0.97	
Event	Disposition	0.82	0.83	
	NoDisposition	0.91	0.95	
	Undetermined	0.56	0.58	
	Overall (micro)	0.87	0.89	0.88
Context	Action	0.54	0.64	0.75
	Temporality	0.64	0.80	0.83
	Certainty	0.69	0.86	0.90
	Actor	0.79	0.97	0.93
	Negation	0.78	0.98	
	Overall (micro)	0.69	0.85	
	Combined	0.42	0.48	

Table 4: F1 scores represent the "micro" F1 scores for the "lenient" metric in the official evaluation script. The "lenient" score reflects some flexibility in the specific character position identified by the model. End-to-end results reflect entire model pipeline, while separate results reflect individual model results. Mahajan et al. [Forthcoming] results reflect their "ClinicalBERT" model from Table 3.

6 Ethics

While beyond the scope of this research, this project raises several ethical issues crucial to the use of NLP in the medical setting:

1. NLP (and computers in general) cannot (and should not) replace a medical practitioner. We are not suggesting that any model replace a human being. Instead, we offer the stated NLP models as a tool for medical practitioners to use to help them fill information gaps, save time, and better care for their patients.

2. The clinical notes used to train our models were written by humans. Every human has some bias (whether it be conscious or subconscious) towards other humans. It is important to note that any bias contained in the training data set will also appear in the models trained on such data sets. This is something to consider not only for this task, but for any learning task that may contain human bias.

7 Conclusion

While our results are comparable to those of the original paper, there is room for improvement. Further error analysis will help pinpoint common errors where our approach fell short. Such errors include where multiple medications are mentioned in the same context window, but refer to different changes. Additionally, some labels in the data set were not seen as often, such as "Negated" in Negation. Imbalanced data can lead to majority label bias, affecting several of our models. Future work that accounts for such imbalance would undoubtedly improve results. In the future, we would also like to consider models other than ClinicalBERT that might be better equipped to handle the different classifications. For example, a Bidirectional LSTM model with the use of gates may have performed better in the Context classification task. Other possible changes could include tuning the size of the context window surrounding a medication mention.

References

E. Alsentzer, J. R. Murphy, W. Boag, W. H. Weng, D. Jin, T. Naumann, and M. McDermott. Publicly available clinical bert embeddings. 2019. doi: arXiv:1904.03323.

A. Kumar, V.and Stubbs, S. Shaw, and Ö. Uzuner. Creation of a new longitudinal corpus of clinical narratives. *Journal of Biomedical Informatics*, 58:S6–S10, 2015. doi: https://doi.org/10.1016/j.jbi.2015.09.018.

D. Mahajan, J. J. Liang, and C. H. Tsou. Toward understanding clinical context of medication change events in clinical narratives. 2020. doi: arXiv:2011.08835.

D. Mahajan, J. J. Liang, and C. H. Tsou. Toward understanding clinical context of medication change events in clinical narratives. Forthcoming. doi: arXiv:2011.08835.