Automatic Triaging of Ophthalmology Patient Phone Calls

Xingchen Liu, Isaac Smith, Julia Truitt, Sophia Wang

Abstract

Patients access their healthcare providers between office visits to address a wide variety of needs, ranging from non-urgent issues such as requesting documents, to urgent issues such as reporting symptoms. Often, telephone calls from patients are triaged by ancillary support staff, technicians, and/or nurses, and the call volumes can be overwhelming. In this project, we develop a system to automatically triage ophthalmology patient phone calls based on electronic health record documentation in the form of free-text notes. We create a specialized ontology for patient phone calls based on description logic which can automatically triage and classify notes into three classes of urgency. We also use several other methods for note classification, including classifiers based on naive bayes and word embeddings, and use these results to refine our ontology. The final ontology was able to infer the urgency classification of 56.4% of the notes with an accuracy and F1 score of 0.539 and 0.627, respectively, compared to independent classification by a medical student which had accuracy and F1 score of 0.616 and 0.465, respectively. Finally, we developed a graphical user interface where a user can input ophthalmology-related text and get an automatic triage classification from the classifiers.

Background

Patients often have many questions and needs that arise between office visits that must be addressed, frequently prompting telephone calls to their healthcare providers' offices.[1–3] In many cases, fielding these phone calls is undertaken by support staff who can be overwhelmed by the sheer volume of inquiries per day in a busy practice,[4] many of which are non-urgent.[5] In the Department of Ophthalmology at Stanford, patient phone calls often initially go through an answering service of technicians and ancillary support staff with minimal ophthalmologic training or background. Messages are documented as notes in the electronic health record and can be forwarded to nurses or technicians with a higher level of ophthalmology training, and those that require a physician's intervention are also forwarded to physicians for review. Providers at every level are sent many messages to address patient needs, and imperfect triaging by providers with varying levels of ophthalmology background can lead to questions that are not addressed in a timely manner.

In this project, we describe the development of a system which can aid in triaging ophthalmology patient inquiries into those that are urgent and should be immediately addressed, those that are semi-urgent and should be addressed within a day, and those that are non-urgent and might be addressed over the next few days. Our approach has been to 1) develop an initial ontology of patient inquiries using expert-based domain-specific knowledge which can triage the urgency of notes documenting patient phone calls using class inference in a description logic framework, 2) use additional natural language processing (NLP) methods of triaging telephone notes, and 3) combine these NLP methods with the ontology to further improve classification, including by refining the ontology using those features found to be especially helpful in triaging patient telephone calls.

Methods

Data Source

We used a relational database that captures data from the Stanford EHR system,[6] which includes clinical narrative text in the health records of individual patients who received care from the Stanford Department of Ophthalmology. We identified all notes longer than 100 characters documenting telephone calls between these patients and the Department of Ophthalmology. We used 100 characters as an arbitrary limit in order to filter out empty notes, as well as to filter out very non-informative extremely short notes, such as "counseling completed." To filter out miscellaneous notes that were documented as telephone calls with patients but were actually not telephone calls, all notes were required to have a match for a regular expression designed to pick up phrases indicating that the "patient called" or "patient calling" for some concern, or some variation thereof. Finally, to enrich for telephone encounters that require some level of responsive action to be taken, a regular expression filter was written to find notes that included some request, such as “please”, “can you”, or similar variations. In all, 1036 notes were included in our corpus, representing 1036 patient phone encounters to be triaged. This project was approved by the Stanford University Institutional Review Board.

Labels

Notes were manually triaged by a board-certified ophthalmologist (SYW) into 3 categories by level of urgency:

- Level 0: Urgent, indicating a concern that should be addressed as soon as possible and within zero days.
- Level 1: Semi-urgent, indicating a concern that should be addressed ideally within one day.
- Level 2: Non-urgent, indicating a concern that could wait to be addressed in 2 days or longer.

Overall, there were 150 notes at urgency level 0, 358 notes at level 1, and 528 notes at level 2.

Representative examples of notes are displayed in Table 1.

Table 1. Example Notes of Different Urgency Levels

Urgency	Example Note
Urgent	patient calling to inform that she is seeing more flashes on her right eye. per patient, the flashes are occuring more constant than what they used to be. patient will like to know if dr, leng will be able to see her today asap? pt is scheduled to see dr. leng tomorrow. please call pt
Urgent	patient calling as she is scheduled for a new visit on 3/24, however, for the past 3 days she has had a curtain/feild cut in vision in her right eye. patient is concerned. requesting a sooner appointment and advice. please return call. (forward to scheduling pool if appropriate)
Semi-Urgent	patient calling to ask dr to pelase fax a refill request for latanoprost (xalatan) 0.005 % drops. per pt, she only has enough for tonight, will like to get this expedited please. pt called her pharmacy but was asked to call md as she only has enough for today.
Semi-Urgent	pt called requesting for a higher dosage of prednisone. pt feels that the dose given is not high enough. she's experiencing more flare ups than usual. please and advise. thanks
Non-Urgent	patient called requesting notes from yesterday appointment with doctor to be faxed to optometrist. please fax
Non-Urgent	patient calling to ask questions about her treatment. is her glaucoma angled or narrow? if narrow why not laser to help instead of surgery?

	what is the status of optical nerve? damaged? if so, how much? what is the average thickness of the nerve? what is the depth of her disc? please return call to advise.

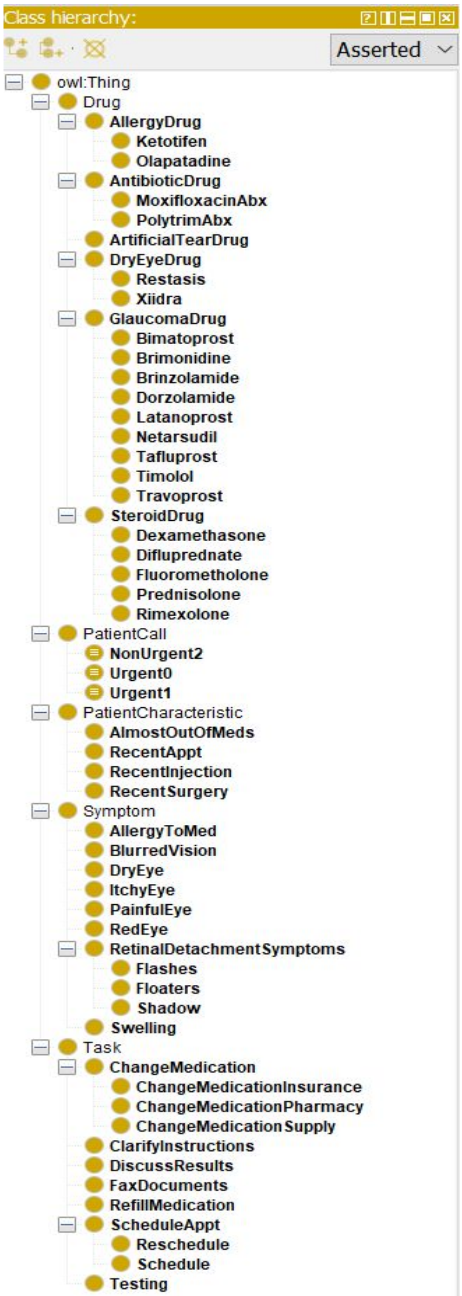

Figure 1. OWL Ontology Classes

Ontology Development

An initial ontology ("naive ontology") to triage patient phone calls was built in the OWL 2 specification using Protege (version 5.5.0)[7] based on domain-specific knowledge priors (SYW). Top level primitive classes describe concepts that might be useful for triaging patient calls, including **Tasks**, which describes tasks that must be accomplished such as changing medication, clarifying instructions, discussing results, scheduling appointments, etc.; **Symptoms** that the patient might be reporting; **PatientCharacteristics**, such as having had a recent surgery; and **Drugs** commonly used in ophthalmology.

Individual words that make reference to the different subclasses of concepts were defined as individuals corresponding to their class. For example, different terms for drugs including generic and branded names were created as individuals belonging to the subclass of that drug. Different words that might be used to describe the **ItchyEye** subclass of **Symptoms**, such as "itchy" or "scratchy," were created as individuals belonging to the **ItchyEye** class.

To facilitate automated reasoning over patient telephone notes to classify them, a **PatientCall** class was created with subclasses of **Urgent0**, **Urgent1**, and **NonUrgent2** corresponding to the different urgency levels of patient phone calls. These urgency subclasses were defined using OWL equivalence statements.

A single object property, *mentions*, was defined in this ontology. This property operates over the domain of **PatientCall**, indicating that a particular patient call *mentions* an individual belonging

to some primitive concept. For example, a **PatientCall** individual might *mention* **pain**, which is an individual of type **PainfulEye**, a subclass of **Symptom**.
After additional NLP methods were developed to analyze and classify the corpus of notes (see below), the OWL ontology was updated ("updated ontology") by adding individuals representing words which may further contribute to note classification by those methods. Additional, minor updates were made to equivalence statements and class structure.

Problem Solving Methods

OWL Ontology

In order to instantiate patient phone call notes as **PatientCall** individuals in the ontology, the ontology was imported into a Python 3 script using the Owlready2 Python API.[8] Each patient note was parsed to determine key words and phrases used, and for each word or phrase corresponding to the name of an ontology individual (e.g. **pain**), a *mentions* relationship was created between the corresponding individual and the newly-created **PatientCall** individual. Once each patient phone call note was added to the ontology as a **PatientCall** individual, the reasoner was applied and the **PatientCall'**s urgency class was inferred.

NLP Methods

Data cleaning

Natural language processing tends to work most effectively when the data is somewhat consistent. Many NLP algorithms invoke some algorithmic method for text processing, such as "stemming" words to make them more consistent, which involves removing prefixes and suffixes, filler words, and other components that don't contribute to the most basic sentence meaning, such as punctuation. Two of our machine learning models vectorized notes based on a "bag of words" method, which, simplistically, considers two sentences more similar if they share the same words in a given vocabulary. This is why words with the same meaning should appear the same, and "words" with little meaning should be ignored.

Call transcripts, conceptually, should contain many of the same types of data, but the actual formatting is very inconsistent, making the comparison of notes to one another--an essential part of NLP--very challenging. Hence, cleaning the data was essential to computationally predicting urgency.

Word2Vec Models

Word2vec is a famous word embedding algorithm whose models are shallow, two-layer neural networks trained to reconstruct linguistic contexts of words. We used a model [9] by Google, which is trained on a 100-billion-word Google News corpus, and used the gensim[10] library to load the model and convert words into 300-length vectors of float values.

For each patient call, we tokenized the notes, ignoring the words that were not present in the word2vec model, and converted the remaining words into unit vectors. The ignored words are mostly medical terms, drugs, or human names and misspelled words or non-words. See the full list at https://pastebin.com/RCZRnwaf.

Each note was then represented as a new vector by simply averaging the unit vectors of the words that occurred in that note. We also built a cache for our vocabulary specifically to speed up the look up, as the original 100-billion-word Google News model was too slow to load. Since each note was converted to its own fixed-length embedding, we were able to use various classification methods to train on the given notes and urgency labels. The methods we used included K-Nearest-Neighbors and Logistic Regression.

To extract the keywords that contributed most to the classification, we came up with a custom method for each classifier. For example for KNN, we defined the keywords as the words with the smallest distance to the averaged note vectors in the top K neighbors whose label was the same as the prediction, among all the words in a note. For Logistic Regression, we used each word as a direct input into the Logistic model, and chose the ones with the highest probability of being predicted as the same label as the whole note.

We also implemented a word whitelist approach, whereby we only extract the word vectors in a particular list, and ignore unmeaningful words such as common words “a/the/of”, connector words like “and/when/while”, etc. After this, we were able to extract more meaning lists of keywords.

Unigram Models

The unigram, or “bag of words”, models we implemented were based on the fundamental concept of converting notes into vectors. These vectors had a length equal to that of our specific vocabulary, and each position referenced a specific word of that vocabulary. After looking at all of the notes in the database, a vocabulary was generated containing all the words seen among all of the notes, in addition to how many times those words appeared. We sorted the words in the vocabulary by how often they appeared over all of the notes, and observed that some words appeared in every note (such as “patient”), while many of the words appeared in smaller subsets of the total note set.

Since it was impractical to use each one of the more than 10,000 “words” appearing among all of the notes when creating our vectors, we chose a subset of words that, when the model was trained, would give us the best predictive accuracy. To do this, we trained 144 separate Support Vector Machines using vectors generated by different subsets of the most frequently used words, varying the length of the vocabulary, and how deeply into the prevalence-ordered vocabulary list we started. We determined that, with this method, the most accurate predictive accuracy of roughly 61% was generated using the 450 most frequently used words among all the notes (Figure 2).

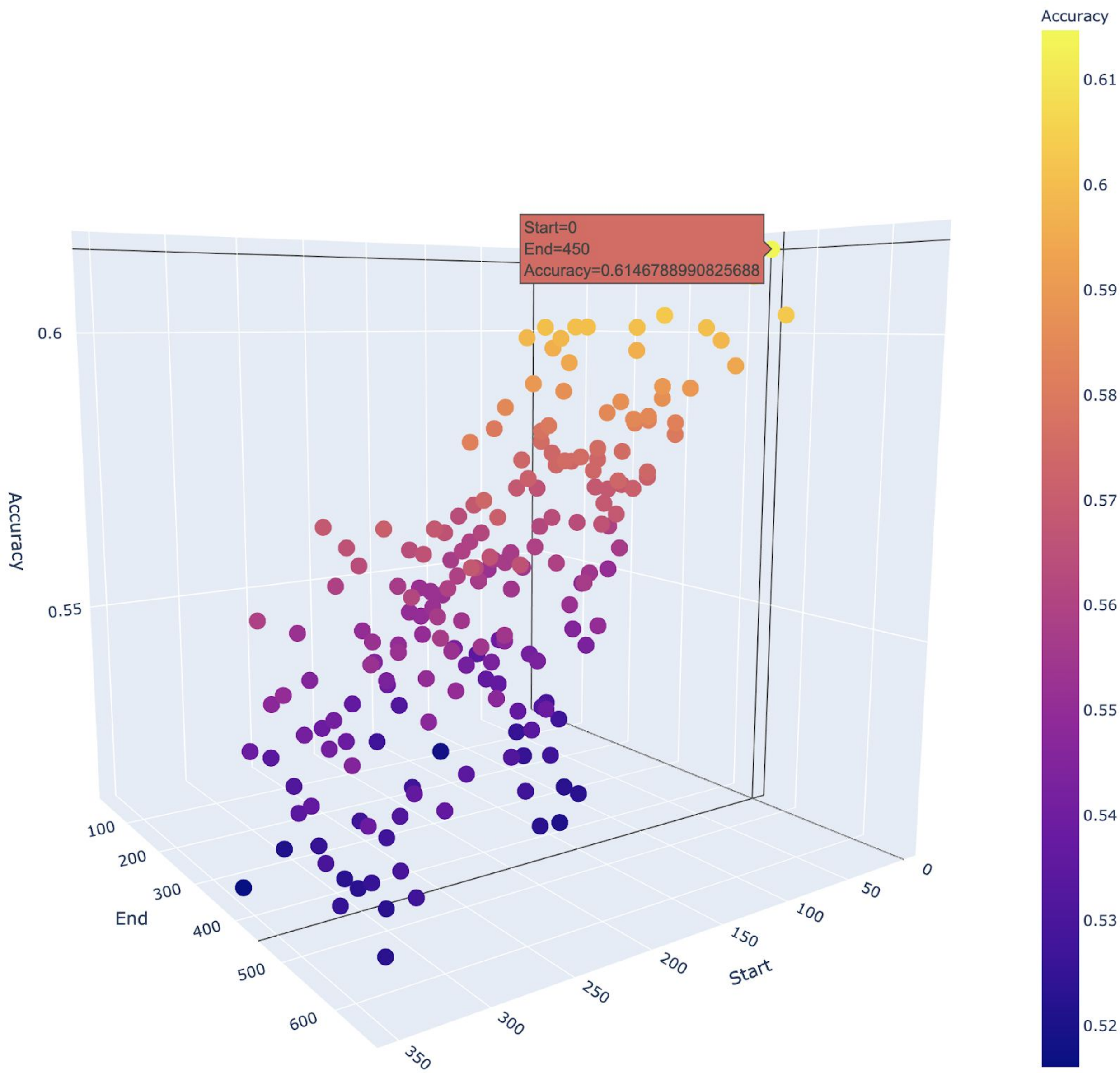

Figure 2

Graphical User Interface

To make the previously described classification techniques more accessible and practical to our target users (i.e. technicians and support staff fielding patient phone calls in a busy ophthalmology practice), we developed a simple web application using the server-side framework Express and the client-side framework React. Through the app, users can directly type in notes from a patient call, press the “submit” button, and view the resulting urgency classification from the selected classifier, as well as a brief explanation in the form of highlighted words that were used to generate the classification.

Evaluation

The performance of our ophthalmology triaging methods was evaluated using standard measures, including accuracy (percentage of predicted labels matching the true labels). Additional standard metrics included sensitivity (recall), specificity, positive predictive value (precision), negative predictive value, and balanced F1 score (2*Precision*Recall/(Precision+Recall)). Because the phone calls were classified into three categories of urgency, these evaluation metrics were calculated in two different ways: once with the two urgent categories (Urgent and Semi-Urgent) grouped together and compared against non-urgent labels, and again with the most urgent category compared against the two less urgent labels (Semi-Urgent and Urgent).

For machine-learning based triaging methods, the final evaluation was performed on a held-out test set of 436 notes. The OWL ontology was evaluated on the entire set of notes.

We also compared the performance of our classifiers to the classifications given by an independent reviewer: a medical student interested in ophthalmology with some clinical experience in ophthalmology. This level of experience may somewhat mimic the background of those non-MD healthcare workers who are often triaging patient complaints in the clinic.

Results

Classification Models

OWL Ontology

When using the Naive Ontology classifier (i.e. the Owlready2 Python script with the original ontology), 719 of the 1036 patient call notes in our database were unable to be classified as one of the three **PatientCall** subclasses. We computed our evaluation metrics for this classifier three times: once completely ignoring all unclassified notes, once using **Urgent0** as the default classification (i.e. any unclassified notes were automatically considered to fit in the **Urgent0** class), and once using **NonUrgent2** as the default classification.

Using **Urgent0** as the default classification is most useful from the perspective of the patient, since the classifier will be less likely to miss a highly urgent note. On the other hand, using **NonUrgent2** as the default classifier may be most useful for busy staffers fielding patient calls, as it would lead the classifier to minimize false positives in terms of urgency, providing staffers with fewer notes to unnecessarily prioritize.

After updating the ontology with additional words discovered to be important from NLP methods (described below), the Updated Ontology classifier (run using the same script, but linking to an updated OWL ontology) was able to classify many more notes than its Naive counterpart; only 452 of the 1036 notes were unable to be classified.

Summary evaluation metrics of the OWL classifiers are shown in Table 2.

Table 2. Results of the Naive and Updated OWL Ontology classifiers.
Numbers in red represent measurements where the Updated Ontology performed worse than the Naive Ontology, and numbers in green represent where the Updated Ontology performed better.

	Ignoring Unclassified Notes		Replacing Unclassified with Urgent0		Replacing Unclassified with NonUrgent2	
	Naive	Updated	Naive	Updated	Naive	Updated
Accuracy	0.612	0.539	0.248	0.352	0.58	0.564
Urgent (0) vs Semi-Urgent/Non-Urgent (1/2)						
F1-Score	0.673	0.627	0.265	0.335	0.511	0.516
Sensitivity	0.77	0.73	0.867	0.82	0.447	0.487
Specificity	0.804	0.876	0.209	0.479	0.949	0.932
PPV	0.598	0.549	0.156	0.21	0.598	0.549
NPV	0.902	0.94	0.265	0.94	0.91	0.915
Urgent/Semi-Urgent (0/1) vs Non-Urgent (2)						
F1-Score	0.802	0.674	0.66	0.626	0.448	0.526
Sensitivity	0.806	0.677	0.925	0.793	0.311	0.433
Specificity	0.669	0.583	0.153	0.286	0.924	0.795
PPV	0.798	0.671	0.513	0.517	0.798	0.671
NPV	0.681	0.59	0.681	0.59	0.582	0.593

Support Vector Machine (SVM) and Gaussian Naive Bayes (GNB) Models

Patient call notes had an irregular, free-text format, and were thus difficult to process computationally in their original state. Using several basic logical filters and regular expressions, we were able to "clean" notes, transforming them from formats like this:

"crm # 123456789 patient called requesting notes from pt. yesterday appointment with doctor to be faxed to optometrist. please fax to 650-123-4567. faxed today. -"

To this:

"patient called request notes from patient yesterday appointment with doctor to be fax to optometrist please fax to fax today"

The first note contains a lot of information, but it's hard to actually standardize this when comparing it to other notes that use slightly different wordings, and some of the information is irrelevant to actual note urgency (like crm #). For example, we don't want to consider "pt." and "patient" or "dr. and "doctor" as different words.

From these "cleaned" notes, we generated our vocabulary and trained both a linear SVM and a Gaussian Naive Bayes model on 600 note vectors with their associated labels. Both models yielded a predictive accuracy that was comparable to that of the word2vec+KNN and the Naive OWL Ontology, but with significantly lower sensitivity and significantly higher specificity. This means that, compared to the other methods, both the SVM and GNB models were prone to misclassifying urgent notes as semi-urgent or non-urgent. In contrast, notes that were marked as being urgent were actually urgent over 96% of the time, and messages marked as either urgent or semi-urgent were marked correctly around 90% of the time (compared to the other methods, which only did this 67%-75% of the time).

We improved these models incrementally by generating a much smaller vocabulary (i.e. more efficient computationally, using only 36 words) that maintained the same computational accuracy of our earlier model, which was based on a vocabulary of 450 of the most commonly occuring words. This vocabulary was generated from the notes with two main goals: coverage and specificity. In this case, coverage meant identifying a subset of words that would cover every note, and specificity meant picking words that had a high variance in their usage among all notes, as well as words that were frequently only associated with certain urgency classifications. With this vocabulary, we streamlined all of the computationally intense stages of the classification problem, while keeping the same level of competency based on our evaluation metrics.

Summary evaluation metrics of the SVM and GNB models are shown in Table 3.

Word2Vec Model

Until the project milestone, the best accuracy we observed was 65% using the Logistic Regression model and 61% using KNN, and the outputted list of keywords used to make urgency classifications did not seem very meaningful, due to the presence of many common/filler words. After the milestone, we continued improving the model by removing common words like “the/and/of”, and also took a whitelist approach by just using a subset of the words in the notes. We manually removed words that have higher frequency and will not help triaging based on common sense, examples are “patient”, “Monday”, “right/left”. We also added SVM as a new classifier based on the word2vec features.

After we implemented the words whitelisting approach (details mentioned in the PSM section above), and added SVM, the outputted list of keywords appeared much more meaningful, catching more expected words like “pain”, “sore”, and “asap”. Additionally, although the accuracy of the SVM model did not improve much (it only went up to 66%) compared to the previous best (LGR, 65%), the specificity went up a bit from 0.952 to 0.968, even though the sensitivity decreased from 0.641 to 0.531.

Future work includes changing to a weighted-average method, and using bio2vec instead of a Google News trained word2vec model.

Summary evaluation metrics of each word2vec model are shown in Table 3.

Summary

Summary evaluation statistics for the above classifiers are shown in Table 3. Characteristics of each classifier are plotted as points along hypothetical ROC curves in Figures 3 and 4, and in Precision-Recall curves in Figures 5 and 6.

Table 3. Summary Evaluation Statistics for Different Approaches to Triage
Note that in this table, the reported evaluation metrics for the Naive and Updated OWL Ontologies ignore any unclassified patient call notes. Green numbers represent metrics where a classifier performed better than the medical student, red numbers represent metrics where the classifier performed worse than the medical student, and bold numbers show the best performance in that row.

Metric / Method	Naive OWL Ontology	Updated OWL Ontology	Gaussian Naive Bayes	SVM	word2vec + Logistic Regression	word2vec + KNN	word2vec + SVM	Medical Student
Overall Accuracy	0.612	0.539	0.571	0.601	0.651	0.61	**0.661**	0.616
Urgent (0) vs Semi-Urgent/Non-Urgent (1/2)								
F1-Score	**0.673**	0.627	0.514	0.556	0.667	0.619	0.618	0.465
Sensitivity	**0.77**	0.73	0.422	0.469	0.641	0.609	0.531	0.333
Specificity	0.804	0.876	0.962	0.962	0.952	0.938	0.968	**0.983**
PPV	0.598	0.549	0.659	0.682	0.695	0.629	0.739	**0.769**
NPV	0.902	**0.94**	0.906	0.913	0.939	0.933	0.923	0.897
Urgent/Semi-Urgent (0/1) vs Non-urgent (2)								
F1-Score	**0.802**	0.674	0.408	0.568	0.708	0.679	0.667	0.717
Sensitivity	**0.806**	0.677	0.278	0.454	0.685	0.685	0.556	0.762
Specificity	0.669	0.583	**0.918**	0.859	0.755	0.673	0.891	0.65
PPV	0.798	0.671	0.769	0.760	0.733	0.673	**0.833**	0.677
NPV	0.681	0.59	0.564	0.616	0.709	0.685	0.671	**0.739**

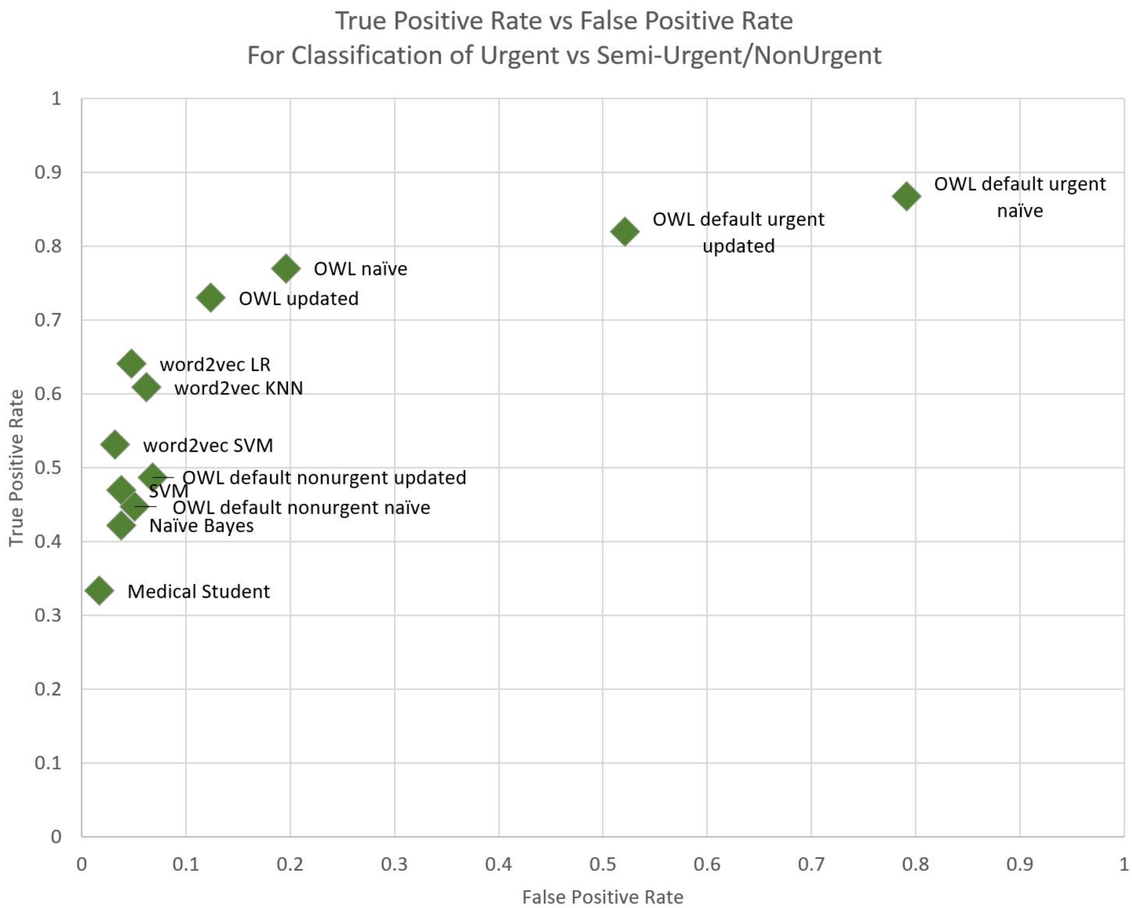

Figure 3. True vs. false positive rate for different classification models, discriminating between Urgent vs. Semi-Urgent/Non-Urgent.

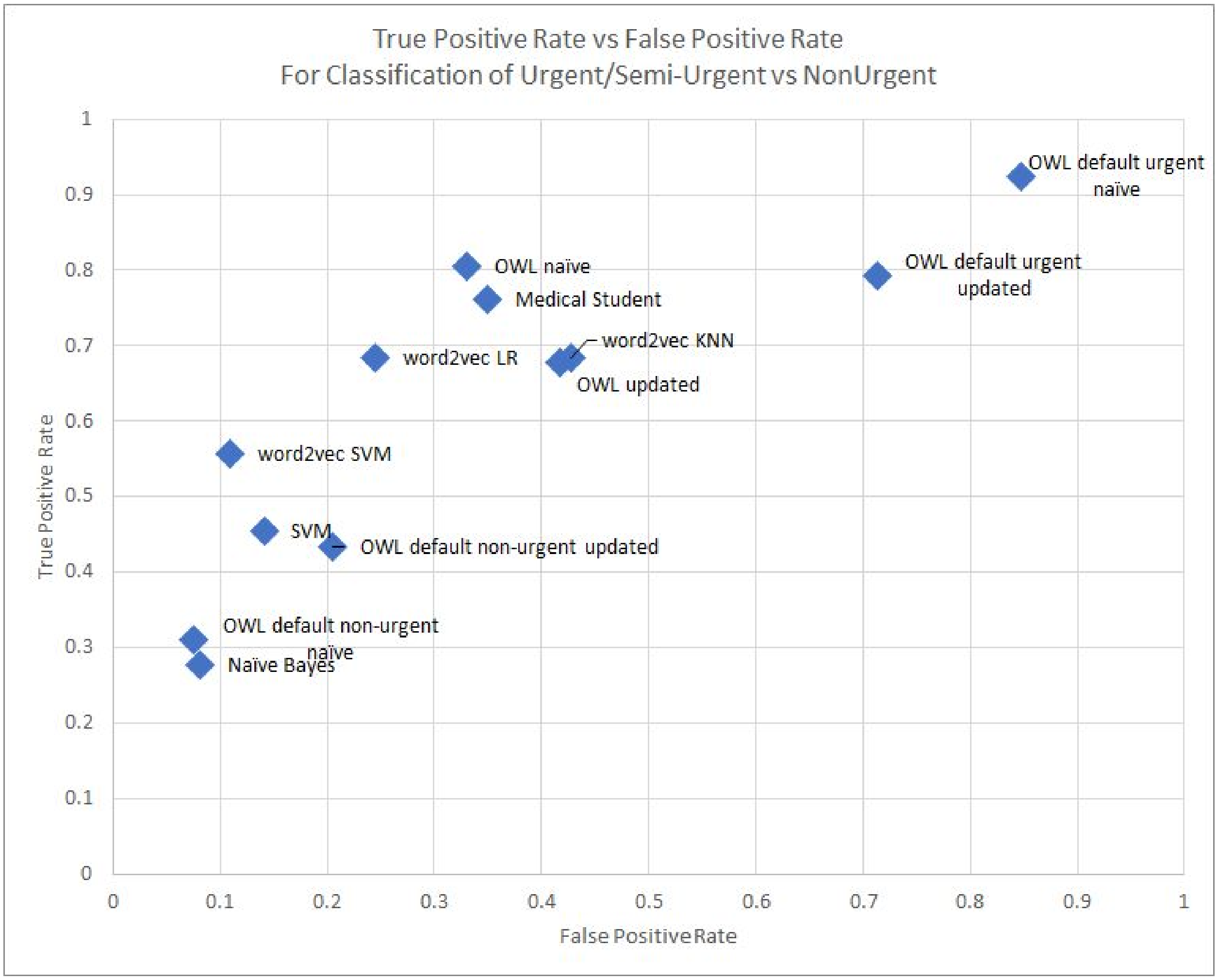

Figure 4. True vs. false positive rate for different classification models, discriminating between Urgent/Semi-Urgent vs. Non-Urgent.

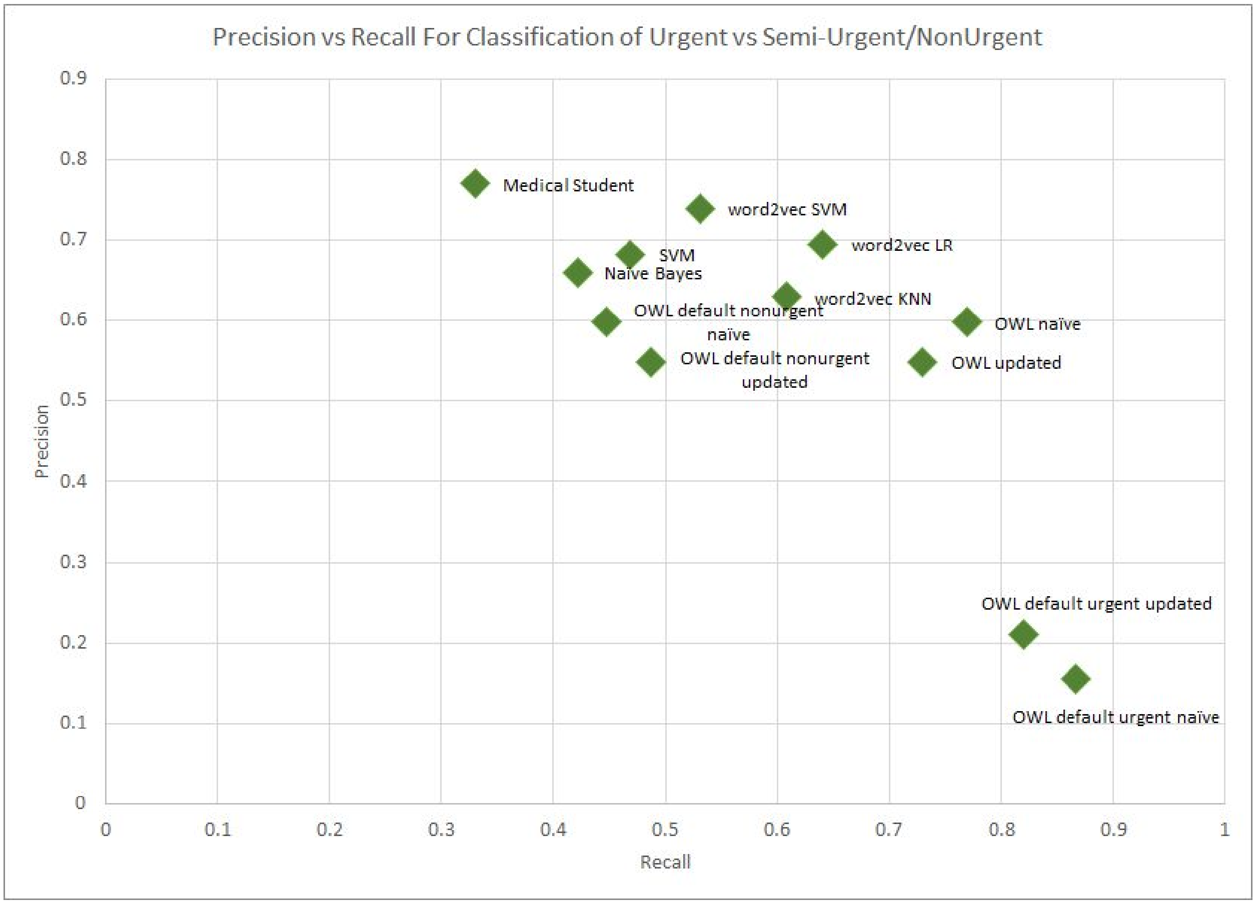

Figure 5. Precision vs. recall for different classification models, discriminating between Urgent vs. Semi-Urgent/Non-Urgent.

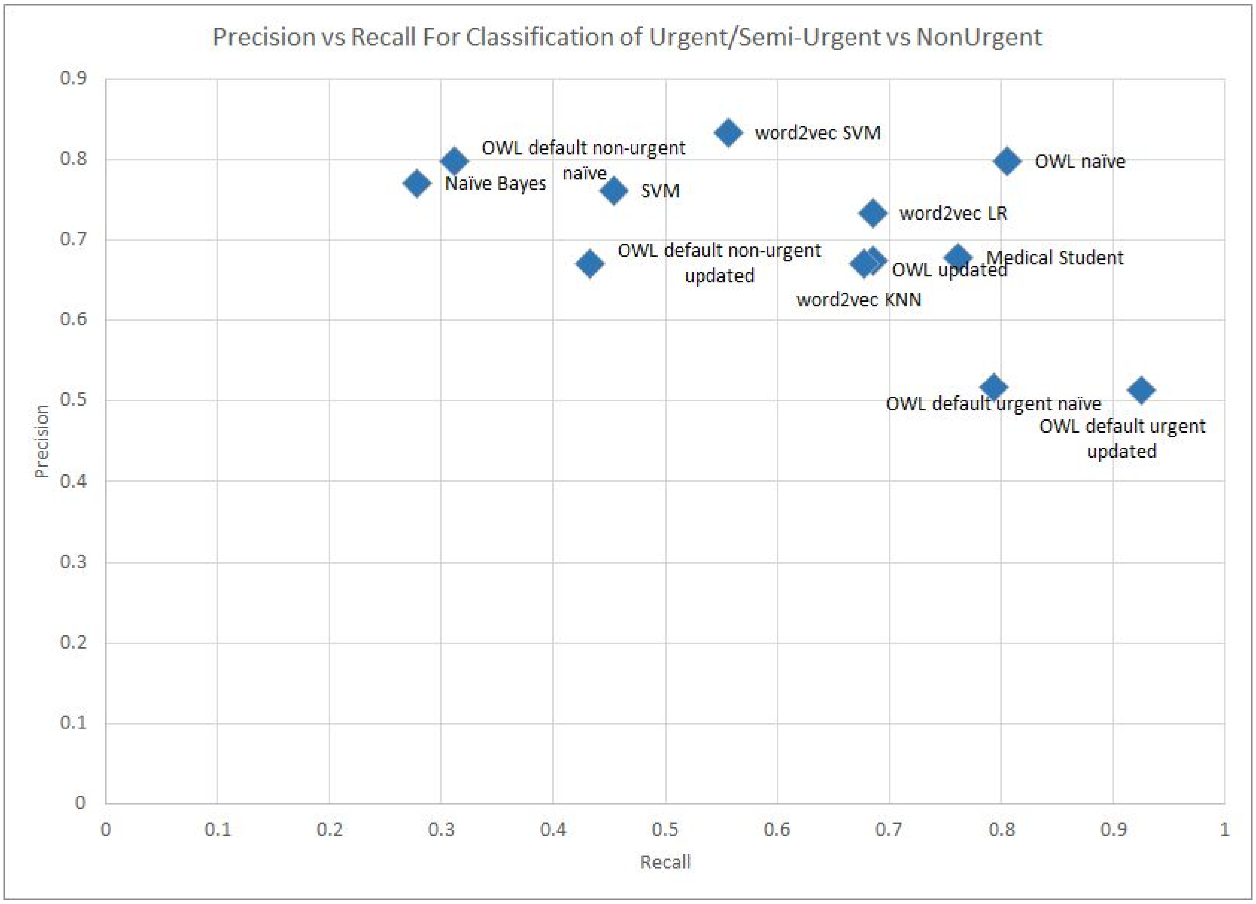

Figure 6. Precision vs. recall for different classification models, discriminating between Urgent/Semi-Urgent vs. Non-Urgent.

Graphical User Interface

A demo of the finalized GUI can be viewed here.

Discussion

In summary, we have developed several approaches towards automatic triaging of ophthalmology telephone notes, including development of an OWL ontology based on domain-specific knowledge, and text-processing methods including those based on word embeddings. We have also developed a preliminary graphical user interface that can take user-inputted text and output a triage classification, as well as highlight the key words in the text which were used to make this classification.

The word-embeddings approach appears to give the highest accuracy. Although the OWL ontology also had comparable accuracy and a higher F1 score, a key limitation was that it was only able to perform inference to classify 31% of the notes initially (up to 56.4% after update), as the equivalence rules defining the urgency classes did not cover all notes, for example in cases when notes did not mention a specific combination of words. However, we found that all of our classifiers were comparable in performance with medical student triage; in other words, a human rater with some experience in ophthalmology did not vastly outperform our classifiers, which demonstrates the potential value of our automated triaging systems.

Our approaches suffer from several limitations which may be areas of further improvement. First, our classifiers do not take into account negation. For example, notes that mentioned "no pain" would not be especially distinguished from notes that describe presence of pain. In particular, description logic is not particularly well-suited to negation, but other NLP methods may be improved in the future to take into account negation. Interestingly, when we updated the ontology with additional important words from the NLP methods, it did not consistently perform better than the naive ontology. This may potentially be related to not taking into account negation when the ontology performs classification based on mentioning words. Although more words in the ontology resulted in many more notes that were classifiable, this did not mean they were necessarily classified more correctly.

Our models also did not account for emotion (i.e., of the patient). One could argue that the patient's emotional state (e.g., irritated, angry) should not impact medical triage of their complaint, but in practice it often does, in that angry patients are often addressed more urgently. On the other hand, if a patient is anxious, in some cases it would be legitimate to have increased concern for their issues. Importantly, our ontology was developed by one physician based on domain knowledge and a subset of patient calls. In reality, an ontology and system of rules would likely need to be developed collaboratively with many participating stakeholders. Finally, implementation and success of triage models depends upon integration into workflow and acceptance by those using it.[11] A future direction may include directly triaging MyHealth patient portal messages which are directly in the patients words, rather than indirectly documented by staff who are handling the phone calls.

In conclusion, we have developed an ontology which can directly use subsumption testing to triage notes from the electronic health record documenting patient telephone calls. The ontology is based on both domain-specific knowledge, as well as natural language processing to determine which are the important keywords. A key limitation of using our ontology to classify notes is that not all notes were able to be classified, due to not mentioning a specific combination of words that fulfill an equivalence statement for a defined urgency class. However, our methods of automatically classifying ophthalmology patient phone calls were comparable to the performance of a medical student triaging the same notes, demonstrating the promise of such automated triaging methods.

Division of Labor

Sophia served as our board-certified ophthalmologist, providing domain expertise and constructing the original ontology based on her clinical knowledge. She also obtained, filtered, and rated the urgency of patient phone call notes, and wrote a Python script to evaluate various training metrics of our numerous classifiers. Isaac wrote the Gaussian Naive Bayes and SVM models. Xingchen wrote all word2vec models. Julia wrote the OWL Ontology classifier and created the graphical user interface.

Acknowledgements

We would like to gratefully acknowledge Amee Azad, the medical student who independently triaged the ophthalmology telephone notes in our corpus.

Authorization to Share

We authorize the use of this project as an example for next year.

Appendix

All code created for this project can be found on our GitHub repo.

References

1. Corral JE, Yarur AJ, Diaz L, Simmons OL, Sussman DA. Cross-sectional analysis of patient phone calls to an inflammatory bowel disease clinic. *Ann Gastroenterol Hepatol* . 2015;28(3):357-365.

2. Jastrowski Mano KE, Gibler RC, Rusy LM, Ladwig RJ, Madormo CO, Hainsworth KR. Seasonal Variation in Pediatric Chronic Pain Clinic Phone Triage Call Volume. *Pain Manag Nurs*. 2017;18(5):288-294.

3. Warren LEG, Kim MB, Martin NE, Shih HA. Analysis of After-Hours Patient Telephone Calls in Two Academic Radiation Oncology Departments: An Opportunity for Improvement in Patient Safety and Quality of Care. *J Oncol Pract*. 2016;12(4):e487-e494.

4. Bowman B, Smith S. Primary Care DirectConnect: How the Marriage of Call Center Technology and the EMR Brought Dramatic Results-A Service Quality Improvement Study. *Perm J*. 2010;14(2):18-24.

5. Greenhouse DL, Probst JC. After-hours telephone calls in a family practice residency: volume, seriousness, and patient satisfaction. *Fam Med*. 1995;27(8):525-530.

6. Data Inventory. STAnford Research Repository (STARR) Tools. https://med.stanford.edu/starr-tools/data-inventory.html. Accessed March 10, 2020.

7. Noy NF, Crubezy M, Fergerson RW, et al. Protégé-2000: an open-source ontology-development and knowledge-acquisition environment. *AMIA Annu Symp Proc*. 2003:953.

8. Welcome to Owlready2's documentation! — Owlready2 0.3 documentation. https://pythonhosted.org/Owlready2/. Accessed March 10, 2020.

9. GoogleNews-vectors-negative300.bin.gz. Google Docs. https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit. Accessed March 10, 2020.

10. Řehůřek R, Sojka P. Gensim: Topic modelling for humans (2017). https://radimrehurek.com/gensim/models/word2vec.html.

11. Bjørn P, Balka E. Health care categories have politics too: Unpacking the managerial agendas of electronic triage systems. In: *ECSCW 2007*. Springer London; 2007:371-390.