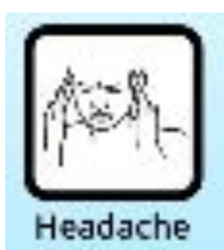
MeDIC (Medication Data to Image Conversion) System

Matthew Ventures and Luke Sturm

Introduction

Prescribing professionals need to explain to their patients how to take their medications. Unfortunately, the main way that this occurs is through their own verbal or written communication[1]. Professionals who work in disaster relief or remote areas can often have trouble communicating to patients how to correctly take medications because of language and cultural barriers. This can put patients at risk. Showing patients pictograms (instructional images) has proven to help address this issue[2][3].



Example Pictogram

We set out to create a mobile application in conjunction with the International Pharmaceutical Federation (FIP) that allows prescribers to take a picture of a medication label and have the relevant pictograms generated for them. The goal of this project is to make taking medications safer for patients and save prescribers time.

Dataset and Features

Image Data: There was no good online dataset of medication labels, so we went into CVS Pharmacy and took multiple images of 43 different common medications.

Terms Data: We search medication labels for matches to a set of 119 different terms that were provided by the FIP. The terms are broken down into five different categories: dose route, frequency, indications, precautions, and side effects. Each term has an associated pictogram image.

Mobile Application

We designed a mobile application in Unity to make prescribing drugs safer for patients. Prescribers can use our app to make medication labels with pictograms instead of text. The prescribers can then give patients these pictogram labels to supplement the text labels. The pictogram labels will help the patient remember when and how much of their medications to take, as well as certain precautions and potential side effects.

Models

Optical Character Recognition:
The first step of our application is translating images of medication labels into text that can be analyzed. We used Google's Optical Character Recognition (OCR) algorithm to accomplish this. This OCR algorithm was implemented using an LSTM neural network that was focused on line detection instead of simply character recognition.

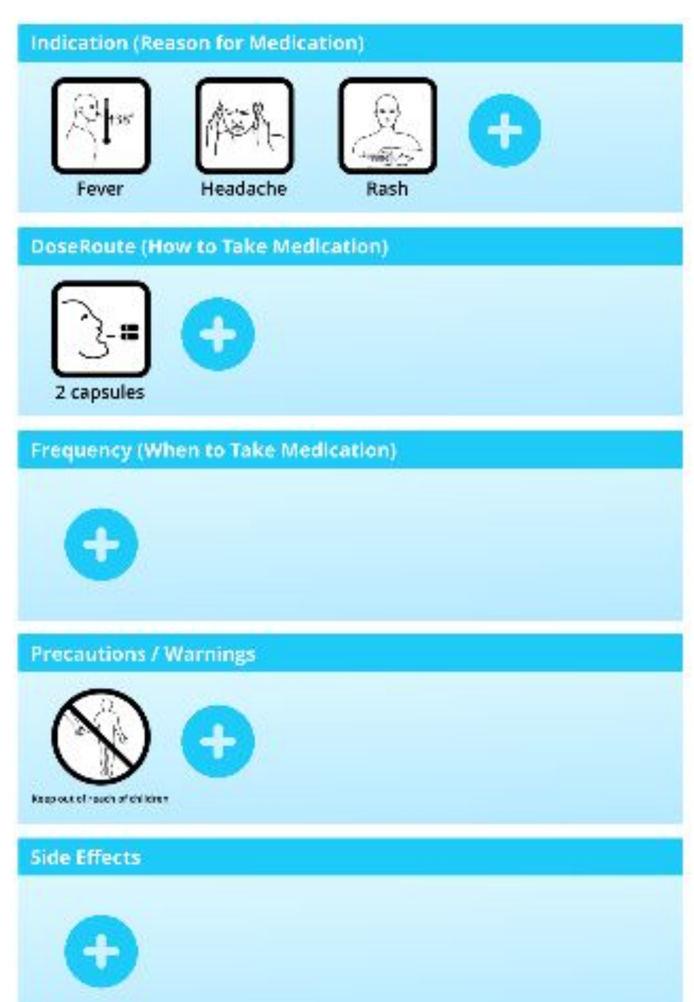
Substring Search:
Our baseline model was a substring search, where we searched the medication text for a direct match of all the terms.

Word Vector Search:
Instead of using a substring search, we implemented a word vector search using GloVe's pre-trained word vectors. We calculated the average word vector of each term and then did an n-gram search through the text (n is the number of words in the term). For each n-gram, we calculated the average word vector. We included the term if we found at least a 90% cosine similarity score between the term and any n-gram.

Improved Substring Search:
We implemented an improved substring search that handled more edge cases. We did this by implementing a synonym translator and a way to normalize frequencies ("4 to 6 hours" becomes "4 hours").

Results

An example of what the app outputs after running our program is shown below. The terms that were found in the search were added to the screen in the correct category. If the prescriber wants to add more terms, they can. By auto-populating these terms, we achieve our goal of saving the prescriber time.



Example app output screen

We compared the outputs of all of our medications to an "Oracle's" output. Below is the summarized performance of all three of our search algorithms. Improved Substring Search was the best of the three. Word Vector Search suffered from a high amount of false positives because average cosine similarity scores didn't capture enough meaning from the text.

	Precision	Recall
Substring Search	0.58	0.67
Word Vector Search	0.31	0.47
Improved Substring Search	0.55	0.77

Results for search algorithms

Discussion

Substring search was able to perform well because the language found on medication labels is fairly standardized. Adding in a synonym translator and accounting for other standard edge cases drove up the model accuracy, precision, and recall.

Using word vectors ultimately failed to provide much value. The pre-trained word vectors meanings didn't correspond to what we wanted. For example, "1" and "5" are very similar according to GloVe, but taking "1 capsule" is extremely different from taking "5 capsules." Also words we wanted to be similar like "caplet" and "capsule" were not similar according to GloVe. Ultimately, word vectors caused too many false positives and were less effective compared to synonyms.

Future Directions

Word Vectors: Ultimately, pre-trained word vectors were not very useful, but we could try to train our own on a medical dataset. BERT word vectors might outperform GloVe word vectors because BERT takes into account context.

Context Detection: Labels are usually split up into different sections: warnings, side effects, etc. We want to incorporate these contexts into our model.

Image Rotation: We want to rotate the image taken to the correct orientation automatically.

References

[1] Mansoor, L. E., & Dowse, R. (2003). Effect of Pictograms on Readability of Patient Information Materials. The Annals of Pharmacotherapy, 37, 1003-1009.

[2] Dowse, R., & Ehlers, M. (2004). Medicine labels incorporating pictograms: do they influence understanding and adherence? Patient Education and Counseling, 58, 63-70.

[3] Yin, H. S., Dreyer, B. P., Schaick, L. V., Foltin, G. L., Dinglas, C., & Mendelsohn, A. L. (2008). Randomized Controlled Trial of a Pictogram-Based Intervention to Reduce Liquid Medication Dosing Errors and Improve Adherence Among Caregivers of Young Children. Arch Pediatr Adolesc Med, 162(9), 814-822.