Text Mining and Predictive Models for Knitting Instructions

Dylan Ross, Applied Data Analysis, Wesleyan University

Acknowledgements: Yuxuan Mei, Hazel Quantitative Analysis Center

Introduction

- Knitting patterns are a rich and underexplored source of structured domain knowledge: they encode relationships between shape, structure, technique, and material.
- Ravelry is the largest knitting website on the internet, composed of tens of thousands of community contributed creations in several languages.
- In this project we develop a natural language processing pipeline to classify a given knitting pattern's construction methods, trained and tested on a preprocessed corpus of 11,840 Ravelry patterns.

Research Questions

- Can a predictive model accurately assign construction attributes given a pattern's provided notes?
- Would the created pipeline be applicable to other areas, such as predicting other attributes (i.e., toy vs garment) or predicting attributes based off the pattern's associated pdfs?

Methods

Sample

- 24,795 Ravelry patterns were collected as .json files, then filtered to only leave patterns that satisfied the following conditions: exclusively English, 100+ characters, and tagged with at least one construction attribute. This left **11,840 patterns** (48% of the original).
- Each Ravelry .json file is formatted as a dictionary, holding hundreds of keys, all assigned a value.

Feature Design

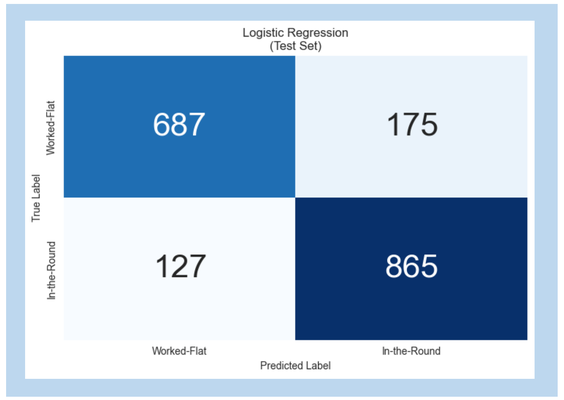
- The most important key for this project is the **"notes"** key. This holds the description written by the author, though it doesn't contain what is detailed in an attached pdf (if there exists one).
- An excerpt from a pattern's notes: "Notes from the designer, Flor:\r\n\r\nWhile the directions for this cap may appear complicated at first glance, it is really an easy hat to knit."
- The other essential piece attribute is the **"pattern_attributes"** array. This is an array of dictionaries consisting of every attribute the author decided to include. These attributes are offered by the website itself, so typos aren't of concern. For this project, we mainly focus on attributes related to construction (list sourced by Ravelry) that appear >=100 times throughout the database.
- Not listed in each .json file are the associated pdfs. These were separately scraped, then paired with each pattern we had access to.

Results

"in-the-round" vs "worked-flat"

- The above construction attributes are the most common across all patterns, so testing began with these two, temporarily trimming the sample size for these exclusive patterns (XOR).

Figure 1. Confusion Matrix for Logistic Regression Model on "in-the-round" vs "worked-flat"

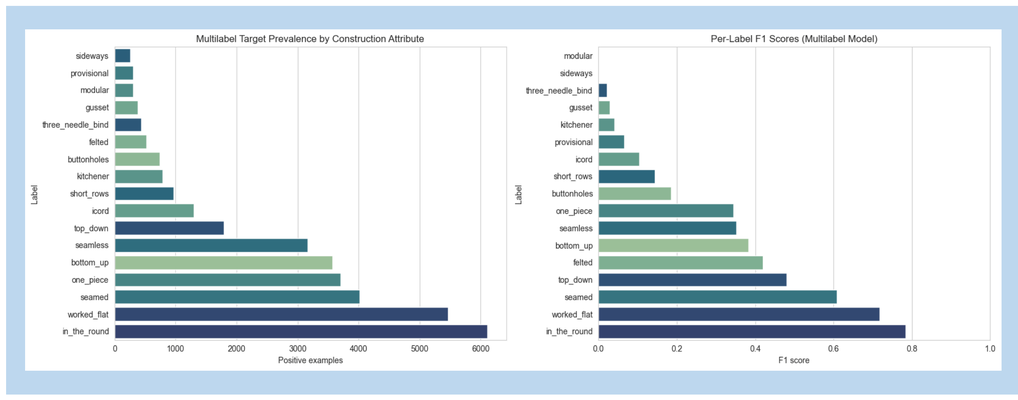


- The logistic regression model on the test set yielded an **accuracy of .8371**, with an **F1 score of .8514** (Figure 1).
- A Multinomial Naïve Bayes model yielded similar results yet consistently fell short of the logistic regression model, leading us to pursue logistic regression.

Multilabel

- Now that the base logic for the model was set, progress could be made on multilabel classification. The goal of the final model is to predict **every construction attribute** tagged with the pattern.
- Accuracy is calculated by when the model is 100% correct, leading to the worryingly low .1537 result. A better metric for measuring accuracy is the Micro-F1 score, which ended up being **.5381**.
- An important trend to note: the fewer instances of an attribute showing up, the less accurate the model becomes (Figure 2).

Figure 2. Multilabel Target Prevalence vs Per-Label F1 Scores



"toy" vs "garment"

- As a secondary task, we ran the same training procedure for a different type of label: toys and garments. The resulting **accuracy is .9332**, with an **F1 score of .9176**.

PDF Text Extraction

- After cleaning each pdf's extracted text and applying the same pipeline, the Micro-F1 score ended up being **.4970**, only .0411 less than the notes model. This proves that our pipeline is applicable to instructions outside of the train/test set.

Discussion and Future Work

- Our proposed pipeline achieves good performance on construction method predictions from pattern texts. In the future, this pipeline could support a feature on Ravelry to automatically suggest tags to include as knitters and pattern makers upload their pattern information to the site.
- For the automatic suggestion feature, we can reformulate the multi-label prediction to output probabilities for each construction method tag, and display suggestions sorted by probabilities.
- For further improvements, we could experiment with more complicated architecture for the classifier, such as neural networks with attention and sequence-to-sequence transformers and comparing with end-to-end LLM based approaches to see the tradeoff between performance and model complexity and computation cost.