Synthetic Biology Laboratory Workflow Constructor

Chris Krenz[1]

Boston University, Boston, MA, USA[1]

Introduction

In biological research and manufacturing, biologists often submit orders to wet labs for various functions, such as DNA synthesis or sequencing. This interface between the client and service provider can be fraught with inefficiencies and barriers to entry. To simplify this process and facilitate the client's ability to submit complex bio-design and bio-manufacturing orders, I am designing a machine learning model to take in natural language prompts to predict the user's desired workflow (i.e. sequence of lab services).

Background

The DAMP Lab at Boston University offers a manageable number of services (and service bundles) from which the model can choose. The training corpus can be acquired through targeted internet searches for these service keywords to find the terminology and phrases that tend to accompany them (for example, references to 'gene editing' or 'modifying sequences' may tend to accompany gene editing services, such as Gibson Assembly and Modular Cloning).

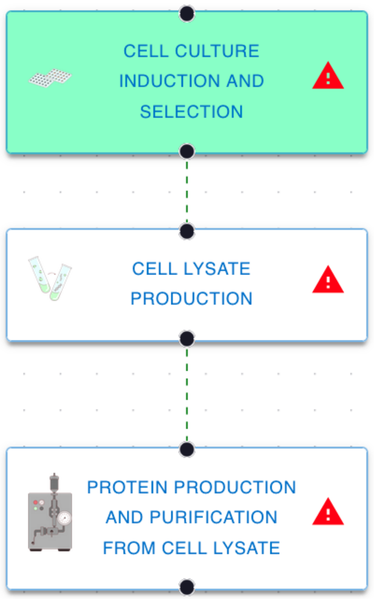

Figure 1: An example workflow in the DAMP Lab ordering system.

Goal

My goal is to develop a model that can translate a natural language description of a desired synthetic biology workflow into an actual sequence of specific laboratory services in order to improve the customer ordering experience.

Methods

The labels for this project were gathered from the DAMP Lab website's listing of available services. While biology labs more broadly can perform a wider range of services, this common set of services will constitute a more manageable number. From this source, I (manually) extracted 38 labels, such as "protein purification" and "gel electrophoresis". The model will attempt to construct a sequence of these services (i.e. a workflow) based on a natural language description of the desired customer work order (roughly 1-3 sentences).

I assembled the training corpus (the input sentences that relate to each of these services) from PubMed. Articles on PubMed are numerous and relatively easy to access via Pythons' BioPython library, specifically, the NCBI Entrez module, which is an API that allows pulling of abstracts from PubMed.

From this source, I fetched 9,999 articles (that contain one or more of the keywords/labels identified above), of which, 9,942 had available abstracts. I further decomposed these abstracts into sentences that contain at least one of the keywords/labels, resulting in 2,687 sentences in the training corpus. For some final pre-processing steps, I removed extraneous characters, converted all words to lowercase to simplify training, removed keywords from sentences to avoid overfitting and enhance contextual learning.

A resulting example sentence would be:

*"plasma sample collected normal healthy individual lung cancer patient **protein purification** analysis conducted using lcmsms"*

Model Details

This model uses a Logistic Regression classifier integrated with TF-IDF (Term Frequency-Inverse Document Frequency) feature extraction to perform multiclass text classification.

1. TF-IDF Vectorization & Feature Extracts

- IDF: Down-weights terms that appear frequently across multiple documents.
- Transforms corpus into a high-dimensional sparse matrix where each entry reflects the importance of a term in a document relative to the entire corpus.
- $\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \log\left(\frac{N}{\text{DF}(t)}\right)$ where t is a term, d is a document, N is the total number of documents, and DF(t) is the number of documents containing term t .

2. Logistic Regression Classifier

- Models the probability that a given input vector $\mathbf{x}$ belongs to a particular class y.
- For binary classification, modeled as: $P(y = 1|\mathbf{x}) = 1/(1 + e^{-(\mathbf{w}^\top \mathbf{x}+b)})$ where $\mathbf{w}$ is the weight vector, and b is the bias term.
- Utilizing a One-vs-Rest (OvR) strategy, separate binary classifiers are trained for each class against all others.
- The model parameters $\mathbf{w}$ and b are optimized using Maximum Likelihood Estimation (MLE), typically via Gradient Descent or Stochastic Gradient Descent (SGD).

3. Training and Evaluation

- Classifier learns to associate TF-IDF features with their corresponding labels by minimizing the loss function, often the Logistic Loss.
- Performance is assessed using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix to ensure balanced and reliable classification across all classes.

4. Mitigating Overfitting

- Applying L2 regularization ($\lambda \sum w_i^2$) penalizes large weights, promoting simpler models that generalize better to unseen data.
- Limiting the number of TF-IDF features and removing high-frequency keywords further reduces the risk of the model overfitting.

Experiments & Results

The primary metric I am using is the f1-score, which balances precision and recall. The model achieved an overall accuracy of 68%, with strong performance in classes like 'Gel Electrophoresis' (f1-score: 0.78). However, certain classes such as 'Protein Purification' exhibited poor performance, likely due to overlapping terminology (e.g. 'peptide' associated with both 'Protein Purification' and 'Gel Electrophoresis').

Overall Accuracy: 0.6821
Classification Report:

	precision	recall	f1-score
Cell Transformation	1.00	0.11	0.20
DNA Extraction	0.67	0.94	0.78
Gel Electrophoresis	0.68	0.91	0.78
Modular Cloning	1.00	0.33	0.50
qPCR Assay	0.80	0.21	0.33

Figure 2: Results for the first 5 labels, showing an overall accuracy of 68% and the specific precision, recall, and f1-scores for each label.

As a specific example, feeding the model the following phrases:

"high separating power immunoblotting synthetic membrane detection peptide"

...correctly yields the label of 'Gel Electrophoresis'. All of these concepts relate to gel electrophoresis (e.g. 'separating power' is the ability of the gel matrix to separate molecules based on size), making this the correct classification.

Conclusion

This model effectively translates natural language descriptions into specific synthetic biology services with an accuracy of 68%. It successfully identifies key lab services, such as DNA Extraction and Gel Electrophoresis but has challenges with overlapping categories. Next steps will focus on expanding the training dataset, refining keyword mappings, and building on the model to accept multiple sentences and return a sequence of services (i.e. a workflow). With these improvements, the model could become a more reliable tool for facilitating efficient and accurate workflow construction in biological research and manufacturing.