An Automated Approach to Terms of Service (TOS) Analysis

Calvin Wang & Manik Mukherjee

Faculty Advisor:
Retsef Levi

Company Sponsor:
Kyle Zeberlein
Zak Raicik

Context & Scope

- Corvus Insurance: leading provider of **commercial insurance products** built on **advanced data science**
- Our project pertains to Technology Errors & Omissions (TEO) insurance, accounting for **1/3** of Corvus services
- Multiple factors within underwriting decision process: cyber infrastructure, history of claims, etc.
- Among them, one important piece is **Terms-of-Service (TOS)** document

Corvus Values & Objectives

Increase Data Sources

Understand the problem

Build a safer world

Problem Statement

Corvus Underwriter

Client to quote

Underwriter receives client name (Top-Level Domain) → **Manually** Search for TOS document through browsing company website → **Manually** look for and approve 3 related clauses

- *Clause #1* ✓
- *Clause #2* ✓
- *Clause #3* ✗

Current Time of Process: **~3 minutes** on average

Main Goal: Automate TOS Extraction/Analysis Process

Timeline

February - March	April	May	June	July	August
On-Boarding: Data & Equipment Access	Query API Development & Underwriter Meetings	Web-Scraping Development & Data Exploration	Website Classifier & Term Extraction Pipeline	Pipeline Codebase & Front-end App Development	Handover Presentation to Stakeholders

Methodology

Input Client Name (Top-Level Domain) → **URL Query/HTML Retrieval** → List of candidate TOS documents → **Website Classifier** → TOS webpage(s) of client → **Key Term Extraction** → List of key paragraphs reported to underwriter

Preprocessing

Simulate human **search engine query** by online API

Domain Processing: Filter Websites with same top-level domain as input client

HTML Retrieval: selenium web-crawler with **parallel processing** to "read" webpages

HTML Text cleaning: Removing non-related elements (images, links, etc.) and removing stop words for ease of model training and final report on front-end tool

Modeling

INPUT: Webpage HTML → Binary Classification Model → Output: Indicator of TOS/non-TOS

Features

Text vectorization methods (bag-of-words) for HTML text encoding and Truncated SVD for **dimensionality reduction** to find feature vectors

Naïve Rule-based key word search indicators (e.g. whether "terms and conditions" appears in URL/HTML)

Search Rank measuring relevance of query result

Model Training

Experimented with popular classification models:
- Decision Tree
- Random Forest
- **XGBoost** (final choice)

Threshold tuning on separate validation set to maximize **F2 score** with heavy weight on **recall**

Term Extraction

Further parsed and cleaned HTML for paragraph-level analysis

Used **Rule-Based** approach (regular expressions) to extract paragraphs with key terms

Met with underwriters to find additional rules and terms that are associated with the existing key terms

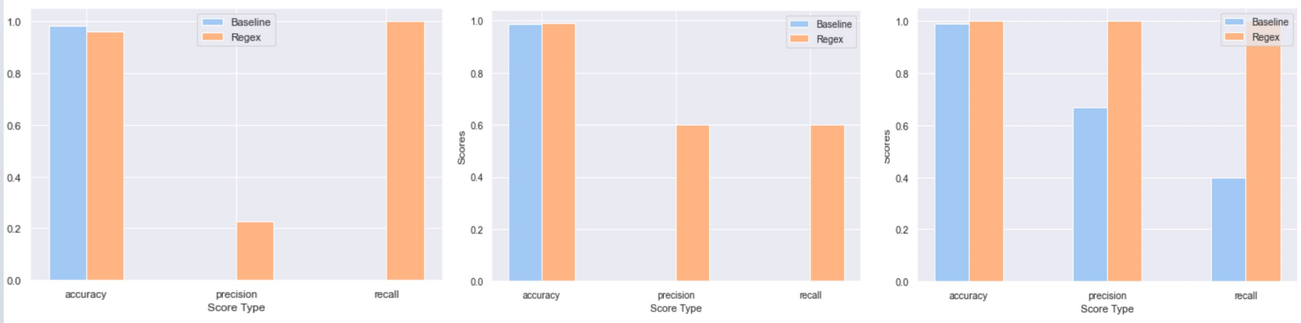
Validation Results

Website Classifier

	precision	recall
Baseline	0.2830	0.8823
Our model	**0.6800**	**1.0**

Outperforms baseline by **+150%** on precision and **+25%** on recall

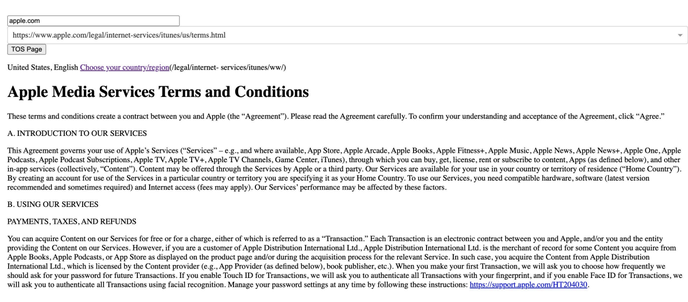
Term Extraction



Our Term Extraction component (in yellow) out-performs baseline method (in blue) in all metrics **in all 3 related clauses**

Average processing time for one client: **15-20 seconds**

Front-End Application



Underwriters input the top-level domain of a client

In case that there are multiple TOS pages, underwriters can navigate to other TOS pages

Color-coding Scheme: paragraphs with different clauses are highlighted in different colors to provide easier visualizations and quicker checks for underwriters.

Business Impact

83%
Reduction in **time per contract**

184
Hours per year saved by underwriters

$20k
Underwriting expense per year saved