An Information Retrieval Engine for All About NLP (AAN)

Thomas George, Irene Li and Dragomir Radev

Department of Computer Science at Yale University and the University of Waterloo

LILY Lab

Introduction

Information retrieval is the process of extracting specific information from a large collection of unstructured documents and resources. This is becoming increasingly relevant in the modern world, where there is an abundance of online data that can get challenging to navigate. Broadly speaking, the objective of All About NLP (AAN) is to supply users with educational resources. In this part of the project, we train a classifier that marks resources as "good" learning materials in the NLP domain, and we extend the model into the CV and STATs domain via transfer learning.

Given the unique problem that AAN aims to solve through, there are no readily available datasets to train the classifier. This presentation will focus on the implementation of the end-to-end software pipeline that was used to scrape the internet and aggregate resources and their features, composing AAN's dataset.

Methods

Referring to *Figure 1*, the pipeline accepts several forms of input. One can supply a set of seeding documents and/or an explicit list of key phrases (examples shown in *Table 1*). The seeding document URIs are forwarded to the DownloadManager that downloads them if they do not already exist in the local filesystem. Then, based on the file type, a parser is created to wrap each seed document. These parsers are forwarded through the '1' channel of the 'Select Mode' demux to a set of 6 deep/statistical key phrase extraction methods (one of which is also specified by user input). Then these selected key phrases from the seeding documents are joined with the user-provided key phrase list.

Moving forward, the signal labelled with '*' carries a 1 because all key phrases of interest are collected. These key phrases are forwarded to a search engine web scraper (selected by the user) which then begins to search for the key phrases and collect the top N URLs for each. These URLs are sent to the DownloadManager, then a parser is built for each consequently downloaded resource. Since signal '*' is now 1, these parsers dump features of their designated document into a csv and the free text into a txt.

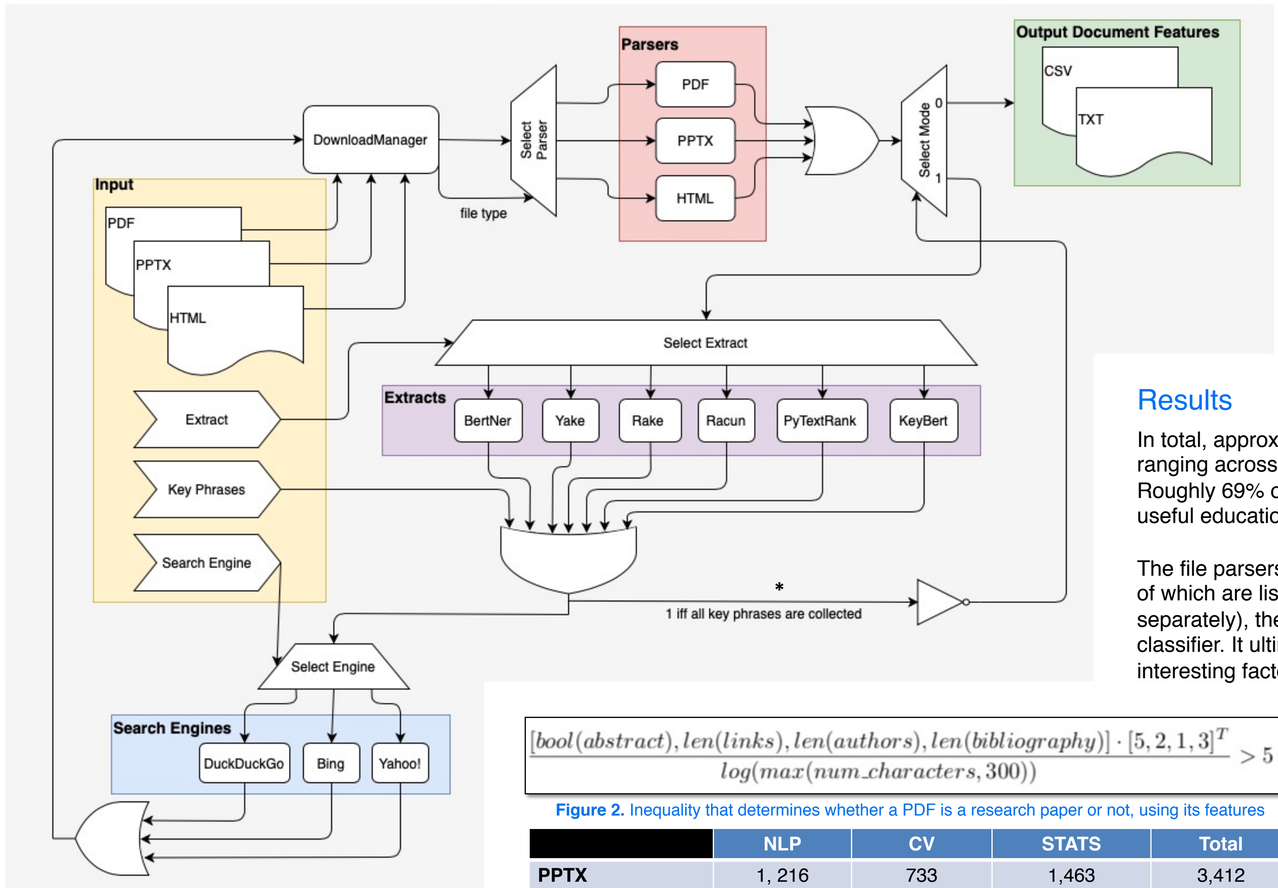

Figure 1. Control Flow of the AAN Data Collection Pipeline

NLP
"word embeddings" site:.edu filetype:.pptx
"text classification tutorial" site:.edu filetype:.pdf
CV
"texture classification" site:.edu filetype:.pdf
STATS
"conditional probability" site:kdnuggets.com filetype:.html

Table 1. Sample key phrase queries in the 3 domains

Implementation Details

One aspect of this pipeline that adds seemingly unneeded complexity is the search engine scrapers. This is because they do not use APIs in order to avoid the financial costs of performing many queries. Rather, they use the selenium and bs4 packages to emulate users clicking through the search engines.

Another interesting facet of this data collection tool is the 3 parsers, which use heuristics to extract details from the document's structure/contents. For example, the PDF parser uses two different libraries depending on whether the document is an academic paper or not. This classification is done using the formula in *Figure 2*.

Results

In total, approximately 40K resources were processed by the pipeline, ranging across the 3 domains and 3 file types (as detailed in *Table 2*). Roughly 69% of those data points were manually annotated as positive (i.e. useful educational resources).

The file parsers extracted 21 distinct features for each document – several of which are listed in *Table 3*. Combined with the 9 deep features (collected separately), the data points held a rich set of features for training the classifier. It ultimately achieved an F1 score of 0.94 and unveiled some interesting factors that characterize an educational resource as "good".

$$\frac{[bool(abstract), len(links), len(authors), len(bibliography)] \cdot [5,2,1,3]^T}{log(max(num\_characters, 300))} > 5$$

Figure 2. Inequality that determines whether a PDF is a research paper or not, using its features

	NLP	CV	STATS	Total
PPTX	1, 216	733	1,463	3,412
PDF	4,961	3,782	1,449	10,192
HTML	9,368	9,302	7,454	26,124
Total	15,545	13,817	10,366	**39,728**
Positive Rate	0.62	0.80	0.65	0.69

Table 2. Control Flow of the AAN Data Collection Pipeline

Group 1		Group 2	
NumHeading	# of headings	NormalizedUnqiueVocab	# of unique words/word
NumEqu	# of equations	PercentTypos	% of words misspelled

Table 3. Sample of features collected. Group 1 features are higher-level (document structure) and Group 2 features are lower-level (natural language structure)

Future Work

It is my goal to continue developing this project in order to further strengthen the data backbone of AAN.

Notably, the key phrase extract, parser construction and web scraping stages can be significantly parallelized. This will increase the iteration speed of adding/improving features and recollecting the data.

Also, more room for improvement can always be found in the document parsers and their heuristics for extracting the structural features of the documents.

Acknowledgement

Thank you to Professor Dragomir Radev for his support and advice on this project and to Irene Li for her guidance.