DataFall: Intelligent Text-based Web Scraping

A more robust way to scrape data from the web

Earl Lee
Yale University
earl.lee@yale.edu
earlvlee@gmail.com

Categories and Subject Descriptors
H.4 [**Information Systems**]: Data Mining

General Terms
Data cleaning, Association rules, Data stream mining

Keywords
NoSQL, web scraping, scraping

1. INTRODUCTION

The past decade has yielded exponential growth of digitized information and, along with it, data driving businesses and individuals in strategic decisions. One study, conducted by the International Data Corporation, cites doubling of information on the Internet at every two years [1]. From a financial perspective, the availability and use of large data sets on the web could generate up to $3 trillion in value within just seven major sectorsâĂŤeducation, transportation, consumer products, consumer finance, electrical, oil and gas, and health care [2]. That valuation is just from the use of open data sets, information that is freely available to the public and normally published by the government.

However, most data-driven action comes from the aggregation of different types of data sets, including open data and date produced by private companies, from a multitude of sources. For example, two former Capital One employees combined stock prices of large retail corporations with credit card transactions to trade stock options, yielding a 1819% return over three years and also fraud investigation by the SEC [3]. The pair extrapolated Chipotle sales volume from credit card transaction data, allowing them to predict earnings calls and purchase options appropriately.

Though the Capital One employees had access to a centralized database with structured data on credit card transactions, other groups, such as the DC-based start-up FiscalNote, must aggregate and process information from a multitude of different sources in order to leverage data in predictive analytics. FiscalNote gathers information such as bill text, bill meta data, legislator voting history, legislator meta data, committee legislator compositions, and more to predict the chances of bills passing.

Data aggregation and processing in this data-driven world generally requires a host of components including including web scraping, entity matching, relational mapping, data standardization, and database population. Furthermore, all this must ideally be done in an online fashion, processing data continuously as it becomes available.

2. PROBLEM

The topic of data aggregation and processing breaks down into numerous different problems, each of which offer a host of different research project opportunities. This paper focuses primarily on web scraping methodologies and the construction of a framework to handle input from different data sources each with its unique format.

Problems pertaining to web scraping are two-fold. First, web scrapers tend to be fragile because of their dependence on the underlying HTML structure of a web page.Traditional web scrapers rely on XPaths and CSS paths in order to extract elements and their associated information from the document object model, also known as DOM, of a web page. However, using these paths to extract data requires balancing robustness and specificity. The more specific a path is, the higher are its chances of failing. For example, a long path that specifically targets one element includes more nodes from the root DOM node to the element's node. Consequently, although there is very little chance of the path grabbing a node that was not intended to be selected, there is a higher chance of the path breaking by a node in the DOM changing, disappearing, or being added in. Some website owners may even be inclined to frequently change the DOM structure in order to make it difficult for external groups to scrape data from one's website, since many view data as currency.

Second, web scrapers generally require high customization for each individual web page. The use of XPaths requires writing unique scrapers for each web page with little code reuse from one page to another. Such individualization also makes large scale web scraping operations tedious to maintain, since the number of potential sources of failures in the operation increase as the number of individual scrapers in-

crease.

On a more broader scale, there is relatively little work done in data aggregation and processing frameworks. Most existing solutions require high levels of configuration making it hard for developers to quickly set up a data processing pipeline for a specific type of data they want to collect.

3. SOLUTION

This project approaches web scraping from an entirely different perspective. Looking at the actual text content of a page to extract data rather than the HTML code. Humans perceive a page by looking at it and making sense of the text content using context clues and visual layout. The ways humans use this information to determine meaning of content can be broken down into heuristics, which are then used to help make sense of text extracted from a page.

On the framework side, the project aims to reduce setup required to aggregate a host of different data sources which contain data on same or similar entities but different pieces of information that must be glued together to form the whole picture. To do so, the project tries matching input data's keys in a JSON object to keys or row names found in a database. When new information that is unlike data presently in the database, schema should be created or modified to accept the dataâĂŤhowever, this has not been fully implemented yet.

4. RELATED WORK

There are numerous web scraping tools out there, including lxml and Beautiful Soup for Python and nokogiri for Ruby. Some web applications also tackle web scraping using visual interfaces, but these sites rely on the same technology as bare bones scraping tools under-the-hood. For scraping frameworks, there is Scrapy for Python, but frameworks are much harder to come by [4].

5. APPROACH OVERVIEW

In order to accomplish text-based web scraping, the project first converts a web page into PDF, which can than be convert into raw text while keeping the structure of the text intact. For example, spacing between text blocks and relative location remains intact. Next, heuristics can be applied on parts of the text that contain desired information to extract such content and make sense of it. The project focuses on understanding information laid out in a tabular form. Next, once data is extracted and held as a JSON object, it can be processed to conform to existing database schema or create or modify schema. Finally, with the data processed to fit nicely with existing data, it can enter the database and undergo validation checks.

Currently, the project code provides proof-of-concept for this approach and uses athletics data on Olympic lifting meet results and athletes to test these concepts. The code consists of four modules, each which serves a different purpose but, together, form the framework.

Clearstream converts input, whether it be a web page or PDF, into raw text while trying to keep as much of the visual structure intact. Dam processes raw and extracts information from raw text. Bridge conforms extracted data into existing schema or modifies existing schema to accept the incoming data. Lastly, Waterfall handles the actual ingestion of processed data and the validation checks that go with it.

5.1 Clearstream: Web to text

The first step to extract data from a page requires converting the web page into text while keeping its structural layout intact. To do this, one can print out a page as a PDF, then convert that PDF into raw text. Using existing libraries, such process can be automated. The source code for this project consists of a host of functions that convert URLs and PDFs into text, including just a URL, a local path, actual PDF data, etc. The availability of these functions encourage users to develop their own processes for converting pages into raw text with layout intact. For example, though this project focuses on web scraping, the tools developed can be used on pure PDF sources.

The wkhtmltopdf (v0.12.2.1) library converts a web page into a PDF. wkhtmltopdf beat out a few other HTML to PDF libraries such as poppler, xhtmltopdf, and weasyprint in preserving the visual layout of a page. wkhtmltopdf, in most cases, would muddle the formatting of a web page, but the changes in layout can be addressed in the post-processing of text.

The downside to converting to raw text is that text styles which made lead to better understanding of textual context are lost in the conversion. For example, large text size could be used to locate identifying information. One way to address the font size issue is by repeating large text across multiple lines, so that text size plays a role in the number of characters a certain phrase takes up in a text file.

5.2 Dam: Processing raw text

Once a page has been has been converted to text, heuristics can be used to discern the meaning of the web page. Using weightlifting meet results as sample data, I was able to take both a weightlifting results PDF and results posted on a website and extract data from them. The results could be used to model two entities in the database, athletes and lifts. Athletes execute six lifts during a meet.

Weightlifting results were presented in a table format. We were able to parse out the table in its pure text format even though the table lacked any borders or other features that may have helped parsing. The reasoning behind making it difficult to parse the table initially is that data is generally presented in a tabular format. Labels are located consistently to the left of a data point. Labels located above a data point generally describes everything that falls underneath it. By cross referencing both labels flanking the top and left areas of a data point, one can normally determine the meaning of the data.

For example, the ESPN player profile page for Tyler Varga, a collegiate athlete, breaks information down using multiple tabular formats.

Parsing a table without borders or formatting to guide parsing of the table serves as a first step to generally apply text

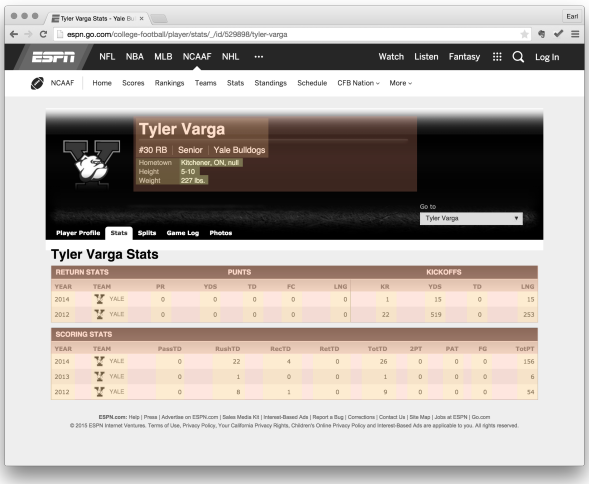


Figure 1: ESPN Player Profile page for Tyler Varga. Highlighting demonstrates breakdown of information in a tabular style.

processing on a page without information formatted using an explicit table.

To parse the table, the header row was extracted, and every character in the row, including whitespace was assigned a corresponding header column. So a character in the 1st cell was assigned to the first header column label while a character in the 17th cell is assigned to the label in the header row closest to that cell. Next, each content row was tokenized into words, with each word being assigned to a label depending on where most of its characters were located. The words for each label were then concatenated and stored as the value to the label. Thus, content row values were paired with header row keys based on the cells they occupied.

```
Name          Class     Result     Rank

Earl Lee       69kg       170kg     1
Bob Smith       69kg     168kg       2
Mark Kim       69kg       132kg       3
-------------------------------------------------
00000000011111111111222222222233333
```

Figure 2: Here, all the cells marked 0 are assigned as 'name'. Cells marked 1 are assigned 'class', and so forth. Thus, content row values can be matched to their appropriate labels.

In the above example table, we may see JSON objects such as the following:

What follows is an actual example from an IWF weightlifting meet result after it was processed and inserted into Mongo:

5.3 Bridge: Fitting data to schemas

Once data is available as a JSON object, it can then be fitted to the database schemas and relations between it and entities existing in the database can be formed. Because a MongoDB database is used to store data, each JSON object

```
{
  "Name" : "Earl Lee",
  "Class" : "69kg",
  "Result" : "170kg",
  "Rank" : 1
}
```

Figure 3: Sample entity extracted from the previous figure.

```
{
  "_id" : ObjectId("55465f..."),
  "born" : "27.04.1993",
  "name" : "LI Yajun",
  "nation" : "CHN"
}
```

Figure 4: Information describing an athlete extracted from a table in weightlifting meet results text.

becomes a document in the database. To conform the object to existing schema, a document collection in Mongo is selected, from which unique keys are extracted. Once a set of keys is obtained from the database, keys in the JSON object are matched to the database keys using string comparison.

Specifically, the open source Python library fuzzywuzzy ranks each database key from order of greatest to least similarity for each key in the JSON object and changes the JSON object's key to the best matching database key if the similarity ranking passes a certain threshold. No database key is used twice, however, so after one database key is matched to a JSON object key, it is removed from the list of possible database keys.

For object keys that have no database key matches with a high enough similarity ranking threshold, they can be added to the schema. Such methodology allows the database to grow as the type and detail of information expands.

```
{
  "Athlete Name" : "Rogers, Martha (M...",
  "Wt. Class" : 63.00,
  "Body Wt." : 63.00,
}

=>

{
  "_id" : ObjectId("55465f43e3784e..."),
  "name" : "Rogers, Martha (Mattie) Ann",
  "class" : 63.00,
  "weight" : 63.00,
  "born" : null
}
```

Figure 5: The first JSON object is an example of one that was scraped from the USAW website, while the second shows what the object gets translated to once it gets inserted into the database.

Next, once the JSON object has conformed to database schema, links can be formed to other documents in the database. To form relations, a target collection and key-value pairs to find documents to form relations with must be provided. The linking happens by searching for possible documents to link to using the collection name and provided key-value pairs. Additionally, a key to identify the relation must be provided, where the value to this key would be ID(s) of linked documents. With this project, athletes were linked to lifts, so one can easily find all the lifts an athlete executed in competition.

For example, here is a lift related to the athlete in the figure above.

```
{
  "_id" : ObjectId("55465f49e3784e0e..."),
  "athlete_id" : ObjectId("55465f43e..."),
  "bodyweight" : "63.00",
  "date" : "2/15/2015",
  "event" : "2015 NATIONAL JUNIOR CHA...",
  "lift" : "total",
  "result" : "201.00"
}
```

Figure 6: The lift is linked to an athlete via the athlete_id key.

5.4 Waterfall: Data insertion

Finally, after molding the data to fit database schema, we must insert it without duplication and managing validation integrity. Though little work has been done during the project in this part of the data processing pipeline so far, Waterfall would be the component for running sanity checks on data and conducting maintenance operation of data. Waterfall would also be the component for preparing data for use by external applications, whether it be implementation of an API to access data or a simple data dump utility.

6. CONCLUSION

This project explores a new approach to web scraping and also a flexible framework for aggregating information from different sources while conforming data to fit database schema. That way, data from multiple sources can be queried and analyzed from a centralized database.

The text-based approach to scraping data from the web proves to be more robust than traditional scraping methods because this approach does not rely so heavily on how the content is formatted. Furthermore, there is much less variation in formatting of content than formatting for HTML code behind the content. Thus, the text-based approach faces less edge cases. However, there are a other issues that arise when relying solely on text-based web scraping. For example, the style of text and other visual cues get lost during the web to text conversion. The conversion can also introduce subtle changes in the layout and format of text, which may diminish information in the text to be analyzed.

On the other hand, there is a lot of work being done in the field of machine learning and natural language processing, which makes text-based approaches to web scraping more powerful. With the implementation of the two fields in text-based web scraping, more automation can occur in identifying information within a text and reduction of possible errors. These methods do require a substantial amount of training data, however, so for sources with little to no data, a heuristics based approach to identifying text on a page will remain king.

7. FURTHER WORK

As the project continues to grow, machine learning and natural language processing modules will be used to break down the information on a page as an alternative to heuristics based approaches. Such an intelligent extraction system will increase in automation and accuracy over time. Furthermore, more work can be done in generalizing and building out the framework for scraping, defining conventions for inputs and adding more tools to extract information formatted in a non-tabular format from a page.

Extracting the desired sections of text within a text source requires more work. For example, a source text file might contain two columns of varying size, however, only the first column might contain useful information while the second column contains advertisements or other extraneous information. Visual algorithms may be able to be used to partition text or even before the text stage, PDFs, into different compartments, which are processed individually.

Lastly, optical character recognition technologies can be utilized to get more accurate textual representation of pages and also extract text from physical documents. By first taking screenshots of a rendered web page and then running OCR on them to get a raw text document, less data may be lost compared to a direct HTML to raw text conversion. Though this would result in more true-to-life text files, it would take a significantly higher amount of processing power.

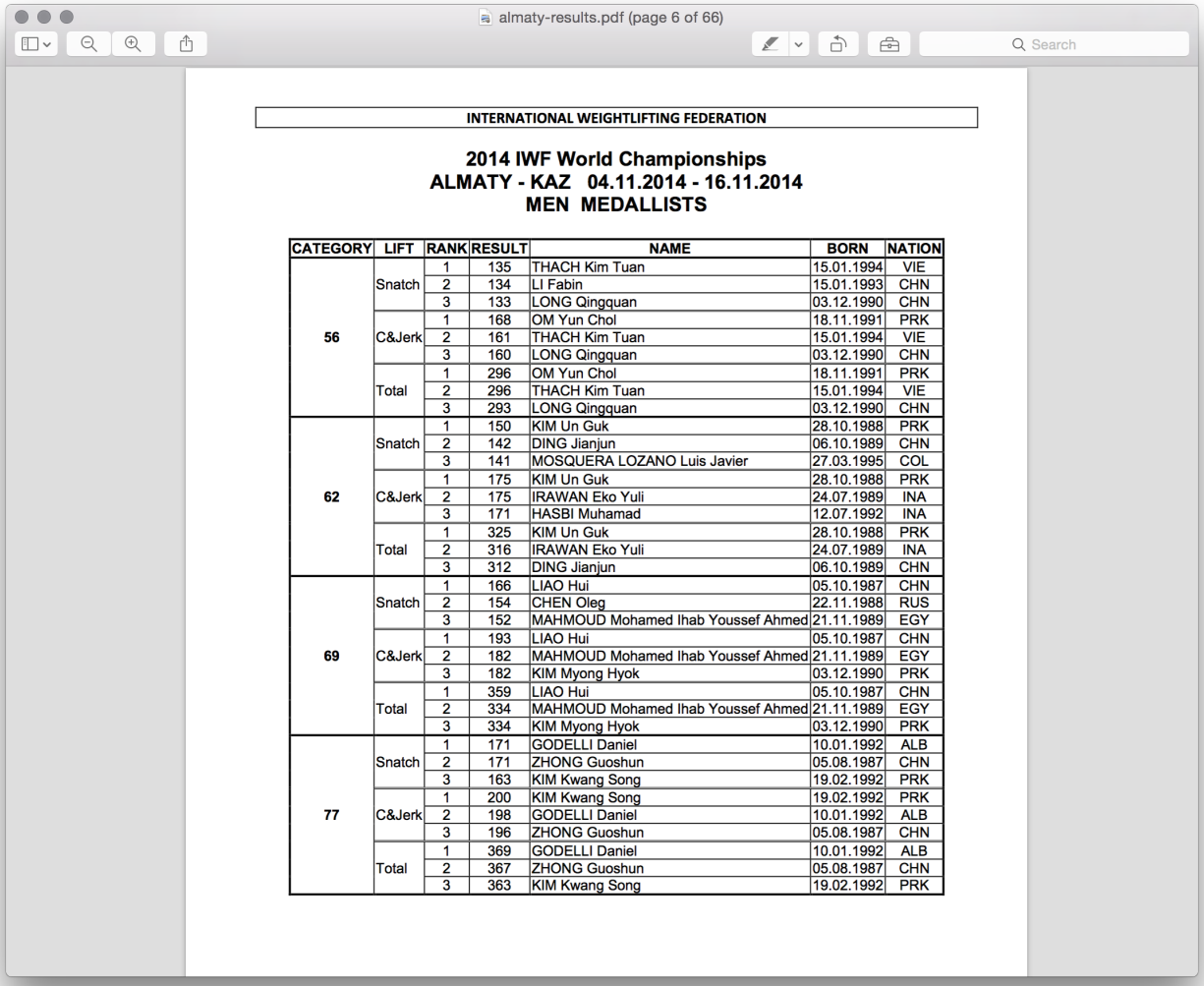
8. REFERENCES

[1] I. D. Corporation. The digital universe of opportunities: Rich data and the increasing value of the internet of things.

[2] D. F. S. V. K. P. G. James Manyika, Michael Chui and E. A. Doshi. Open data: Unlocking innovation and performance with liquid information, 2013.

[3] M. Levine. Capital one fraud researchers may also have done some fraud, 2015.

[4] Scrapy. Scrapy | a fast and powerful scraping and web crawling framework, 2015.



INTERNATIONAL WEIGHTLIFTING FEDERATION

2014 IWF World Championships
ALMATY - KAZ 04.11.2014 - 16.11.2014
MEN MEDALLISTS

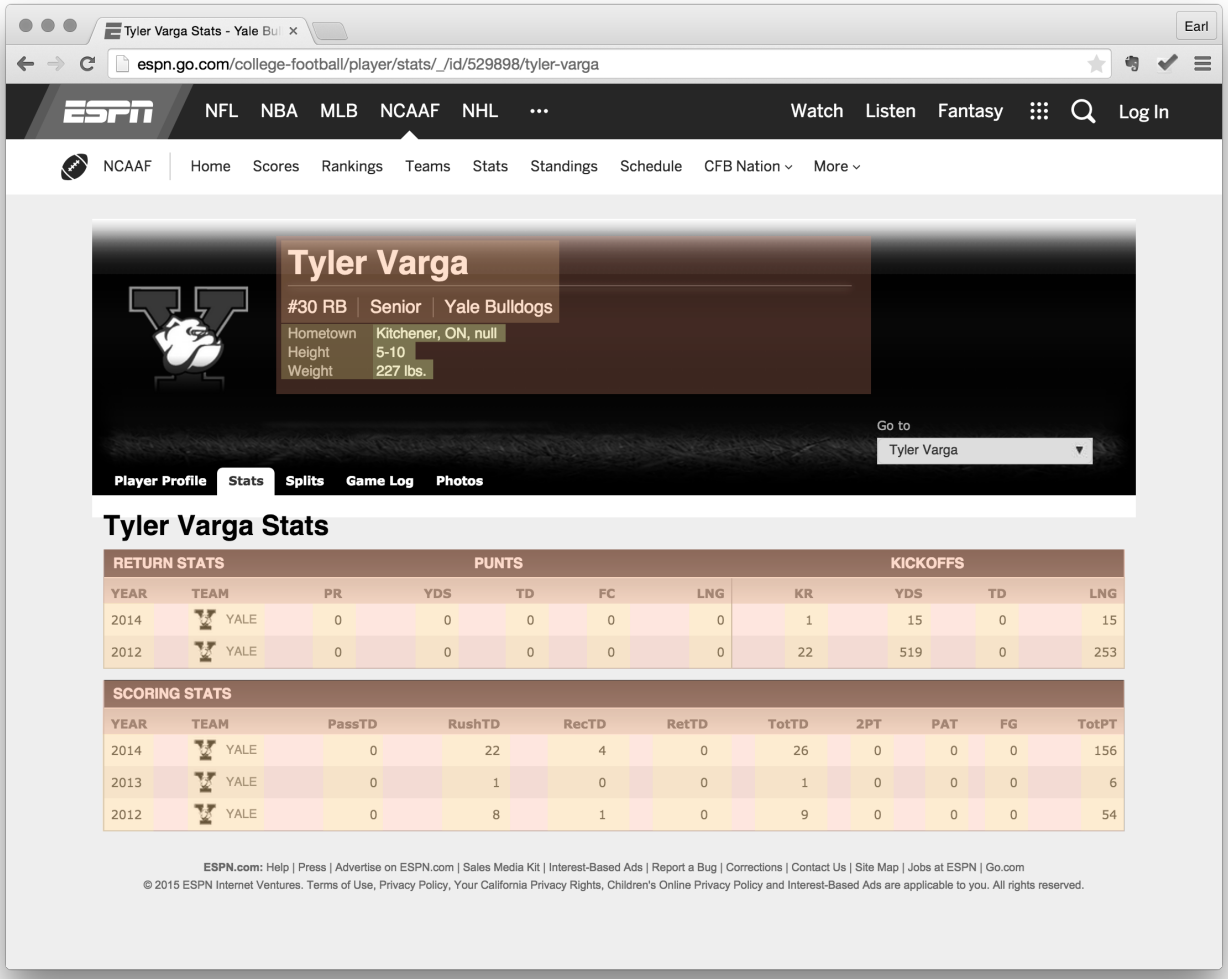
CATEGORY	LIFT	RANK	RESULT	NAME	BORN	NATION
56	Snatch	1	135	THACH Kim Tuan	15.01.1994	VIE
		2	134	LI Fabin	15.01.1993	CHN
		3	133	LONG Qingquan	03.12.1990	CHN
	C&Jerk	1	168	OM Yun Chol	18.11.1991	PRK
		2	161	THACH Kim Tuan	15.01.1994	VIE
		3	160	LONG Qingquan	03.12.1990	CHN
	Total	1	296	OM Yun Chol	18.11.1991	PRK
		2	296	THACH Kim Tuan	15.01.1994	VIE
		3	293	LONG Qingquan	03.12.1990	CHN
62	Snatch	1	150	KIM Un Guk	28.10.1988	PRK
		2	142	DING Jianjun	06.10.1989	CHN
		3	141	MOSQUERA LOZANO Luis Javier	27.03.1995	COL
	C&Jerk	1	175	KIM Un Guk	28.10.1988	PRK
		2	175	IRAWAN Eko Yuli	24.07.1989	INA
		3	171	HASBI Muhamad	12.07.1992	INA
	Total	1	325	KIM Un Guk	28.10.1988	PRK
		2	316	IRAWAN Eko Yuli	24.07.1989	INA
		3	312	DING Jianjun	06.10.1989	CHN
69	Snatch	1	166	LIAO Hui	05.10.1987	CHN
		2	154	CHEN Oleg	22.11.1988	RUS
		3	152	MAHMOUD Mohamed Ihab Youssef Ahmed	21.11.1989	EGY
	C&Jerk	1	193	LIAO Hui	05.10.1987	CHN
		2	182	MAHMOUD Mohamed Ihab Youssef Ahmed	21.11.1989	EGY
		3	182	KIM Myong Hyok	03.12.1990	PRK
	Total	1	359	LIAO Hui	05.10.1987	CHN
		2	334	MAHMOUD Mohamed Ihab Youssef Ahmed	21.11.1989	EGY
		3	334	KIM Myong Hyok	03.12.1990	PRK
77	Snatch	1	171	GODELLI Daniel	10.01.1992	ALB
		2	171	ZHONG Guoshun	05.08.1987	CHN
		3	163	KIM Kwang Song	19.02.1992	PRK
	C&Jerk	1	200	KIM Kwang Song	19.02.1992	PRK
		2	198	GODELLI Daniel	10.01.1992	ALB
		3	196	ZHONG Guoshun	05.08.1987	CHN
	Total	1	369	GODELLI Daniel	10.01.1992	ALB
		2	367	ZHONG Guoshun	05.08.1987	CHN
		3	363	KIM Kwang Song	19.02.1992	PRK

Appendix A: A sample page from the a weightlifting results PDF. This particular PDF was not converted from a webpage but directly obtained from the International Weightlifting Federation website.

2014 IWF World Championships
ALMATY – KAZ 04.11.2014 – 16.11.2014
MEN MEDALLISTS

CATEGORY	LIFT	RANK	RESULT	NAME	BORN	NATION
		1	135	THACH Kim Tuan	15.01.1994	VIE
	Snatch	2	134	LI Fabin	15.01.1993	CHN
		3	133	LONG Qingquan	03.12.1990	CHN
		1	168	OM Yun Chol	18.11.1991	PRK
56	C&Jerk	2	161	THACH Kim Tuan	15.01.1994	VIE
		3	160	LONG Qingquan	03.12.1990	CHN
		1	296	OM Yun Chol	18.11.1991	PRK
	Total	2	296	THACH Kim Tuan	15.01.1994	VIE
		3	293	LONG Qingquan	03.12.1990	CHN
		1	150	KIM Un Guk	28.10.1988	PRK
	Snatch	2	142	DING Jianjun	06.10.1989	CHN
		3	141	MOSQUERA LOZANO Luis Javier	27.03.1995	COL
		1	175	KIM Un Guk	28.10.1988	PRK
62	C&Jerk	2	175	IRAWAN Eko Yuli	24.07.1989	INA
		3	171	HASBI Muhamad	12.07.1992	INA
		1	325	KIM Un Guk	28.10.1988	PRK
	Total	2	316	IRAWAN Eko Yuli	24.07.1989	INA
		3	312	DING Jianjun	06.10.1989	CHN
		1	166	LIAO Hui	05.10.1987	CHN
	Snatch	2	154	CHEN Oleg	22.11.1988	RUS
		3	152	MAHMOUD Mohamed Ihab Youssef Ahmed	21.11.1989	EGY
		1	193	LIAO Hui	05.10.1987	CHN
69	C&Jerk	2	182	MAHMOUD Mohamed Ihab Youssef Ahmed	21.11.1989	EGY
		3	182	KIM Myong Hyok	03.12.1990	PRK
		1	359	LIAO Hui	05.10.1987	CHN
	Total	2	334	MAHMOUD Mohamed Ihab Youssef Ahmed	21.11.1989	EGY
		3	334	KIM Myong Hyok	03.12.1990	PRK
		1	171	GODELLI Daniel	10.01.1992	ALB
	Snatch	2	171	ZHONG Guoshun	05.08.1987	CHN
		3	163	KIM Kwang Song	19.02.1992	PRK
		1	200	KIM Kwang Song	19.02.1992	PRK
77	C&Jerk	2	198	GODELLI Daniel	10.01.1992	ALB
		3	196	ZHONG Guoshun	05.08.1987	CHN
		1	369	GODELLI Daniel	10.01.1992	ALB
	Total	2	367	ZHONG Guoshun	05.08.1987	CHN
		3	363	KIM Kwang Song	19.02.1992	PRK

Appendix B: A sample of the text extracted from the PDF of the weightlifting results in appendix A.



Appendix C: ESPN Player Profile page for Tyler Varga. Highlighting demonstrates breakdown of information in a tabular style. The brightest yellow boxes can be considered to be children of surrounding boxes, and the content within these yellow boxes correspond to a label found in an enclosing box.

Tyler Varga
#30 RB
Senior
Yale Bulldogs

HometownKitchener, ON, null
Height5-10
Weight227 lbs.

Go to

Tyler Varga

Player Profile
Stats
Splits
Game Log
Photos

Tyler Varga Stats

RETURN Stats		PUNTS					KICKOFFS		
YEAR	TEAM	PR	YDS	TD	FC	LNG	KR	YDS	TD
2014	YALE	0	0	0	0	0	1	15	0
2012	YALE	0	0	0	0	0	22	519	0

SCORING Stats								
YEAR	TEAM	PassTD	RushTD	RecTD	RetTD	TotTD	2PT	PAT	FG
2014	YALE	0	22	4	0	26	0	0	0
2013	YALE	0	1	0	0	1	0	0	0
2012	YALE	0	8	1	0	9	0	0	0

Appendix D: ESPN player profile page of Tyler Varga converted into text using web page to PDF to text conversion. Note that the last column was removed due due to line-wrapping.