Technische Universität München

Interdisziplinäres Projekt der Informatik

Design und Implementierung eines XPath Web-Scraping Frameworks mit Anwendung in Fallstudien zur Talentforschung in der Leichtathletik und im Fußball

In Kooperation mit dem Lehrstuhl für

Trainingswissenschaft und Sportinformatik

Sebastian Hofstetter

Aufgabensteller: Prof. Dr. Martin Lames

Betreuer: Prof. Dr. Martin Lames

I. Inhalt

1. Motivation

Data-Mining und Machine-Learning erlauben mit steigenden Rechnerkapazitäten immer mächtigere Auswertungen von Datenbeständen. Voraussetzung ist eine qualitativ möglichst hochwertige Datenbasis, die aus mehreren Datenquellen zusammengefügt sein kann. Oft liegen die Quellen jedoch nicht in maschinell verarbeitbarer Form vor. Da eine händische Konvertierung und Aufbereitung der Daten oft zu teuer ist und nicht skaliert, kommt regelmäßig sogenanntes Data-Scraping beziehungsweise Web-Scraping im Internet zum Einsatz. Hierfür gibt es bereits eine Vielzahl an Werkzeugen[1,2,3], die aber kostenpflichtig sind oder Programmierkenntnisse voraussetzen.

Ziel dieser Arbeit soll eine Lösung sein, welche minimale Vorkenntnisse voraussetzt, indem eine graphische Benutzerschnittstelle durch die Konfigurationsschritte führt. Soweit möglich sollen Auswahlfelder die Einstellung erleichtern und übrige Informationen vom Programm selbst heuristisch ermittelt werden. Projekte sollen weitgehend wiederverwendbar sein, so dass Beziehungen zwischen bereits gesammelten Daten hergestellt werden können. Um ein *skalierbares* System zu schaffen, das in kleineren Fällen trotzdem *effizient* arbeitet, werden zwei spezialisierte Implementierungen bereitgestellt. Weitere Anforderungen finden sich im Kapitel *Anforderungen und Spezifikationen* dieser Dokumentation.

Beim Web-Scraping muss der Rechtslage besonderes Augenmerk zukommen. Grundsätzliche unterliegen veröffentlichte Werke dem Urheberrecht. Hiervon ausgenommen sind Fakten sowie Werke niederer Schöpfungshöhe[4]. Besondere Rechte können gegebenenfalls den Allgemeinen Geschäftsbedingungen der Webseitenbetreiber entnommen werden. Diese wurden in der Vergangenheit jedoch von Gerichten zurückgewiesen, wie zum Beispiel im populären Urteil des Landgerichts Hamburg gegen die Fluggesellschaft Ryanair, das einem Vergleichsportal explizit die kommerzielle automatisierte Auswertung der Webseite gestattete[5]. Generell sollte sich mit der Thematik bereits vor Erstellung der Web-Scraping Aufgaben auseinandergesetzt werden.

[1] scrapy (http://scrapy.org/) ein Web-Scraping Paket für die Programmiersprache Python
[2] Ubot studio (http://ubotstudio.com/index7) eine kostenpflichtige Lösung für Automatisierung von Web-Aufgaben
[3] 80legs (http://www.80legs.com/) ein kostenpflichtiger Web-Scraper und - Crawler
[4] Gesetz über Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz)
[5] Aktenzeichen: 3 U 191/08

2. Anforderungen und Spezifikationen

Im Folgenden werden die Ergebnisse der Anforderungsanalyse sowie die zugehörigen Lösungsansätze dargestellt.

2.1. Plattformunabhängigkeit

Plattformunabhängigkeit ist wünschenswert, da das System des Anwenders nicht unmittelbar bekannt ist, beziehungsweise flexibel sein soll.
Aus diesem Grund wird eine Client-Server Lösung gewählt. Der Dienst wird als Webapplikation zur Verfügung gestellt, den ein Client mit beliebigem Browser wahrnehmen kann. Der Server wird in Python implementiert, das von allen gängigen Betriebssystemen unterstützt wird. Weiterhin wird bei der Auswahl der eingesetzten Frameworks und Pakete darauf geachtet, dass auch diese für alle Plattformen zur Verfügung stehen.

2.2. Skalierbarkeit

Die Menge der zu verarbeitenden Daten kann erheblich variieren. Große Datenmengen setzen hohe Ansprüche an die Serverleistung und das dahinterstehende Datenbanksystem. Skalierbarkeit kommt jedoch nicht ohne Nachteile. Ein System, das mit der Knotenzahl eines Clusters skaliert, ist im kleinen Betrieb in der Regel deutlich weniger effizient, als ein für den Ein-Maschinen-Betrieb ausgelegtes System. Da sich diese Pfade bis heute nicht vereinen lassen, werden auch zwei Pfade bei der Implementierung gewählt.
Für das hochskalierbare System kommt *web2py*[6] als Web Framework zum Einsatz, da es den Instanzen wenig Overhead aufbürdet und bereits ein Grundgerüst mitbringt (Scaffolding App). Es unterstützt als Plattform Google App Engine[7], das als Serverinfrastruktur und NoSQL Datenbanksystem gewählt wird. Auf diese Weise können kleine Anwendungsszenarien auf einem eigenen Entwicklungsserver oder kostenlos auf Google Servern laufen. Für sehr große Anwendungsszenarien können beliebig viele Zusatzinstanzen gemietet werden. In der Implementierung wird aus diesem Grund auf einen hohen Grad an parallelisierbaren Code geachtet.
Im Gegensatz dazu kann das *hocheffiziente* System die meisten Anwendungen auf einem einzelnen privaten Server leisten. Als Web Framework kommt *django*[8] zum Einsatz, so dass mittels ORM[9] eine große Auswahl an Datenbanksystemen angebunden werden können. Für kleinere Projekte ist das

[6] http://web2py.com/
[7] https://cloud.google.com/products/app-engine/
[8] https://www.djangoproject.com/
[9] Object Relational Mapper: Abstraktionsschicht zwischen objektorientiertem Code und einem Datenbanksystem

mitgelieferte SQLite aber ausreichend. Alternativ könnte beispielsweise das kostenlose und sehr leistungsfähige PostgreSQL konfiguriert werden.

2.3. Assistenten der Benutzerschnittstelle

Für die Benutzung soll ein Minimum an Programmiervorkenntnissen und Einarbeitungs- beziehungsweise Pflegeaufwand benötigt werden. Aus diesem Grund wird eine graphische Benutzerschnittstelle gewählt, die mittels Auswahlfeldern durch den Erstellungsprozess einer Web-Scraping Aufgabe führt.

2.3.1. Spezifikation der Quelldaten

Für die Spezifikation der Quelldaten kommen intern *XPath* und Reguläre Ausdrücke (*RegEx*) zum Einsatz, die für diesen Einsatzzweck standardisiert wurden. Reguläre Ausdrücke können, aber müssen nicht, vom Anwender genutzt werden, da sie automatisch aus einer Auswahl an Datentypen abgeleitet werden. Beispielsweise werden für die Extraktion von Fließkommazahlen aus einem Fließtext nur kompatible Stellen mittels des Ausdrucks `\d[\d.,]*` untersucht. Für die Generierung der XPath Ausdrücke zur Auswahl des Datenvorkommens können weiterhin browserinterne Hilfsmittel genutzt werden (siehe Kapitel 4). Zur unmittelbaren Validierung der Selektoren kann zudem ein spezieller Testmodus verwendet werden.

2.3.2. Datentypkonvertierungen

Daten werden automatisch in ein standardisiertes Format konvertiert. Beispielsweise werden international variierende Zeit-, Datums- und Zahlenformate automatisch erkannt. So wird sowohl „10.000,00“, als auch „10,000.00“ und „February, 11th 2011“ wie auch „11.02.2011“ unterstützt. Die Werte werden automatisch aus Fließtexten extrahiert und umschließende Leerzeichen (*Whitespaces*) entfernt.

2.3.3. Web-Crawling

Komplexe Aufgaben können rekursiv beschrieben werden. Beispielsweise stellen viele Webseiten ihre Daten seitenweise mit einem Link auf die nächste Seite dar. Eine rekursive Aufgabe kann sich selbst als Quelle für die Generierung dynamischer URLs sowie als Ziel für die extrahierte Daten enthalten. Der so entstehende Auftrag arbeitet sich analog zu einem Web-Crawler selbstständig durch eine Webseite. Die Auswahlfelder für komplexe Aufgaben werden in der Oberfläche zwecks Übersichtlichkeit nur bei Bedarf eingeblendet.

2.3.4. Wiederkehrende Aufgaben

Da sich die Quelldaten regelmäßig ändern, ist es sinnvoll, Web-Scraping Aufgaben mehrmals durchzuführen. Dabei muss eine Aufgabe idempotent arbeiten, da sonst Duplikate entstehen. Hierfür können für jeden Datensatz ein oder mehrere Schlüssel (*Primary Key*) als eindeutige Identifikatoren des Datensatzes spezifiziert werden.

2.4. Zugriffsschutz

Wenn das System auf externen Servern läuft, sollen nur autorisierte Anwender Zugriff auf die extrahierten Daten, die Auswertungen sowie auf die Erstellung neuer Aufgaben haben.
Für diesen Zweck wird eine minimale Benutzerkontensteuerung eingesetzt, die nur vom Administrator bestätigten Anwendern Zugriff gestattet.

2.5. Datenschnittstelle und Export

Die extrahierten Daten müssen in weiteren Prozessen verarbeitet werden können. Sie müssen also in einem standardisierten Format vorliegen und/oder exportierbar sein.

- Auf dem *skalierbaren* System ist dies der Google NoSQL Datastore, der über eine Vielzahl an Schnittstellen angesprochen werden kann[10].
- Auf dem *effizienten* System liegen die Daten in relationalen Formaten vor, abhängig vom gewählten Datenbanksystem. Sie können mit jeder SQL-kompatiblen Anwendung verknüpft und verarbeitet werden.

In jedem Fall existiert zusätzlich eine Exportfunktion nach Excel beziehungsweise CSV, um händische Auswertungen und Aufbereitungen durchführen zu können.

[10] https://cloud.google.com/appengine/docs/python/storage

3. Implementierung

3.1. Implementierungsdetails

	***Skalierbare* Implementierung**	***Effiziente* Implementierung**
Zielsystem	Google Appengine[11] Cluster oder eigener Appscale[12] Cluster	Ein oder wenige WSGI kompatible Server
Datenbanksystem	Google Datastore (NoSQL)[13]	Beliebiges SQL System[14], SQLite vorkonfiguriert
Programmiersprache	Python 2.7.9, HTML 5	Python 3.4.2, HTML 5
Sourcecode	github.com/BastiCambeo/idpscrapper Branch: *web2py-port*	github.com/BastiCambeo/idpscrapper Branch: *django-port*
Web Framework	web2py 2.9.10	django 1.7
Abhängigkeiten	• web2py • google app engine sdk • lxml (html/xml/xpath Support[15]) • requests (http Support[16]) • feedparser (Datumskonvertierungen) • xlwt (Excel Export)	• django • lxml • requests • feedparser • xlsxwriter (Excel 2007 Support) • django-picklefield (NoSQL Support)
Codezeilen (LoC)[17]	Python: 1500 HTML: 400 JavaScript: 300 über 127 Commits	Python: 1800 HTML: 250 JavaScript: 300 über 147 Commits

[11] Google Appengine ist ein Platform as a Service (*PaaS*) Dienst für Webanwendungen: https://cloud.google.com/appengine/docs
[12] Appscale ist eine freie Implementierung des Google Appengine Stacks: http://www.appscale.com/get-started/
[13] https://cloud.google.com/appengine/docs/python/storage
[14] https://docs.djangoproject.com/en/1.7/ref/databases/#
[15] http://lxml.de/tutorial.html
[16] http://docs.python-requests.org/en/latest/
[17] Aktuelle Statistiken finden sich unter https://github.com/BastiCambeo/idpscrapper/graphs/punch-card

3.2. System Design

Beide Implementierungen folgen einer strikten Model-View-Controller Trennung[18]. Die Anbindung der Modelle an die Datenbank kommt in beiden Fällen über einen ORM[19] zustande. Die View wird über ein Client-Server Modell vom Controller getrennt. Die Kommunikation findet über eine *RESTful JSON API* statt.

Eine Web-Scraper-Aufgabe besteht im Datenmodell aus dem Namen und einer Menge von Daten-Selektoren sowie URL-Selektoren. Die Datenselektoren grenzen die Daten einer Seite über einen *XPath Ausdruck*, einen *Regulären Ausdruck* und einen *Datentyp* ein. Ein Daten-Selektor wie beispielsweise SPIELER_ID beschreibt eine Spalte einer Ergebniszeile und kann optional Bestandteil des *Primären Schlüssels* sein. Ein URL-Selektor beschreibt eine Seite, auf der der Web-Scraper arbeitet, und kann in einer dynamischen URL optional Elemente aus bereits gesammelten Ergebnissen enthalten.

Die Aufgaben werden auf dem *skalierbaren* System asynchron über die Taskqueue API[20] abgehandelt. Die einzelnen Worker arbeiten hier auf einer beliebigen Anzahl an Serverinstanzen.

Auf dem *effizienten* System werden die Aufgaben hingegen synchron auf einer einzigen Instanz ausgeführt.

Nachdem der Quelltext einer Seite angefordert wurde, werden die Selektoren in einem Präprozessorschritt angewendet. Zuletzt findet eine heuristische Konvertierung in den Zieldatentyp der Spalte statt. Auf diese Weise können auf der Datenbank später Datentypspezifische Abfragen ausgeführt werden, wie beispielsweise eine Sortierung der Ergebnisse nach Wochentagen.

3.3. Tests

Die Benutzerschnittstelle wurde unter Mozilla Firefox 37, Google Chrome 40 sowie dem Microsoft Internet Explorer 11 getestet. Der serverseitige Code wird über automatisierte Unit-Tests (DocTests) und Integrations-Tests abgedeckt.

[18] Unter Django heißen die *Controller* traditionsgemäß *Views* und die *Views* sind als *Templates* benannt
[19] Object Relational Mapper: Abstraktionsschicht zwischen objektorientiertem Code und einem Datenbanksystem
[20] https://cloud.google.com/appengine/docs/python/taskqueue/

4. Bedienanweisung

4.1. Installation des Servers

Zunächst wird der aktuelle Sourcecode mittels *git* in ein beliebiges Verzeichnis heruntergeladen.

```
git clone https://github.com/BastiCambeo/idpscrapper.git
```

Abhängig von der gewählten Implementierung unterscheidet sich das weitere Vorgehen leicht. Für den ersten Versuch wird ausdrücklich die Verwendung des *effizienten* Systems empfohlen.

4.1.1. Skalierbares System

Nur für sehr große Projekte empfohlen. Die Implementierung wird über den Branch ausgewählt.

```
git checkout web2py-port
```

Es muss Python 2.7.9[21] sowie das Python Google Appengine SDK[22] heruntergeladen und installiert werden. Die Paketabhängigkeiten sowie *web2py* sind bereits enthalten.
Nun lässt sich der Entwicklungsserver starten,

```
/path/to/google_appengine/dev_appserver.py path/to/sourcecode
```

beziehungsweise das Projekt auf einen kostenlose Google Appengine Server deployen.[23]

```
/path/to/google_appengine/appcfg.py --oauth2 update path/to/sourcecode
```

4.1.2. Effizientes System

Die Implementierung wird über den Branch ausgewählt.

```
git checkout django-port
```

Es muss Python 3.4.2[24] heruntergeladen und installiert werden. Nun können die Packetabhängigkeiten nachgeladen werden.

```
easy_install django
easy_install lxml
easy_install requests
easy_install feedparser
easy_install xlsxwriter
easy_install django-picklefield
```

Für Windows existieren alternativ vorbereitete Installationspakete.[25]
Einmalig müssen die Datenbankrelationen angelegt werden.

```
python path/to/sourcecode/manage.py migrate
```

Der Entwicklungsserver mit *SQLite* Datenbanksystem lässt sich nun starten.

```
python path/to/sourcecode/manage.py runserver
```

Als Server lässt sich alternativ Apache über die WSGI Schnittstelle verwenden.[26]
Auch kann das Datenbanksystem gegen beliebige SQL Systeme ausgetauscht werden.[27]

[21] https://www.python.org/download/releases/2.7.9/
[22] https://cloud.google.com/appengine/downloads
[23] https://cloud.google.com/appengine/training/go-plus-appengine/deploy
[24] https://www.python.org/download/releases/3.4.2/
[25] http://www.lfd.uci.edu/~gohlke/pythonlibs/
[26] https://docs.djangoproject.com/en/1.7/howto/deployment/wsgi/modwsgi/
[27] https://docs.djangoproject.com/en/1.7/ref/databases/

4.2. Bedienung des Clients

Nun kann die Benutzerschnittstelle über die konfigurierte lokale URL und Port im Browser geöffnet werden: http://127.0.0.1:8080

Die im Rahmen des Interdisziplinären Projekts erstellten Aufgaben können über folgende URL geladen werden: http://127.0.0.1:8080/idpscraper/put_tasks

Nun sollte sich die folgende Seite präsentieren, wobei die Abbildungen bei Verwendung der *skalierbaren* Version leicht abweichen können.

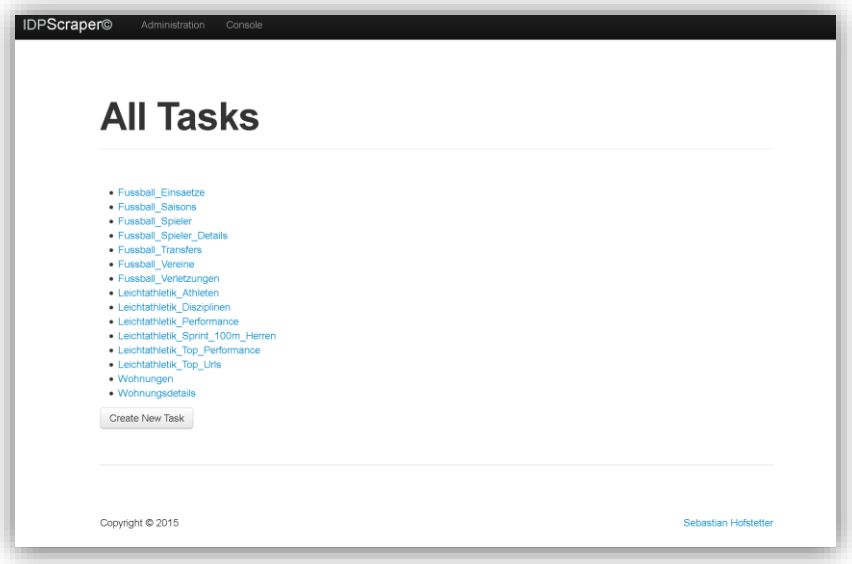

Abb. 1 Aufgabenverwaltung

4.2.1. Erstellung einer Web-Scraping Aufgabe

Mittels der Aufgabenverwaltung lässt sich eine neue Aufgabe durch Benutzung des entsprechenden Buttons erstellen. Danach gelangt man in die vereinfachte Detailansicht einer Web-Scraping-Aufgabe.

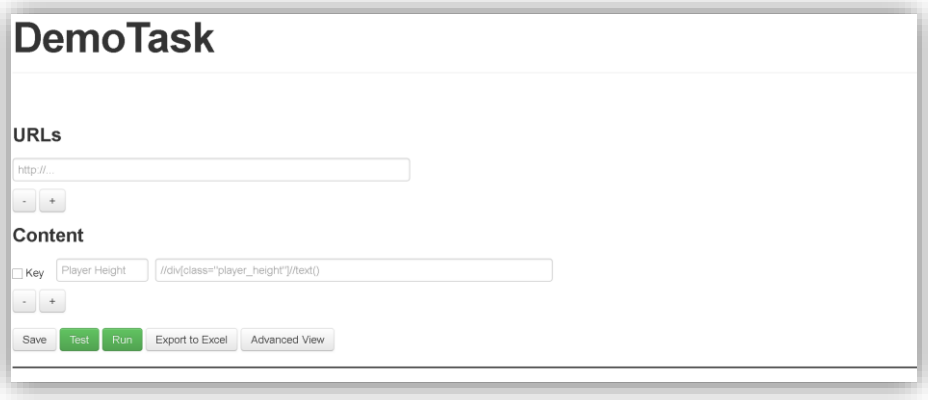

Abb. 2 Vereinfachte Detailansicht einer Web-Scraping Aufgabe

In dieser Ansicht finden sich die URL-Selektoren und Daten-Selektoren. Ersteres ist eine Liste von URLs, die bearbeitet werden soll. Die Daten-Selektoren beschreiben die Spalten einer Ergebniszeile. Neben dem Spaltennamen ist mindestens ein XPath Ausdruck zur Spezifikation der Quelle innerhalb der Seite nötig. Dieser kann einfach mit Hilfe einer Browser Erweiterung ermittelt werden.[28] Das Ziel des XPath Ausdrucks müssen alle vorkommen der gewünschten Daten auf der spezifizierten Webseite sein.

Wenn die Checkbox *Key* ausgewählt wird, werden die Daten dieser Spalte Bestandteil des *Primären Schlüssels* des Datensatzes. Dies ist nötig, um Idempotenz bei mehrfacher Ausführung der Aufgabe zu gewährleisten.

Weitere Optionen lassen sich in der erweiterten Ansicht einstellen.

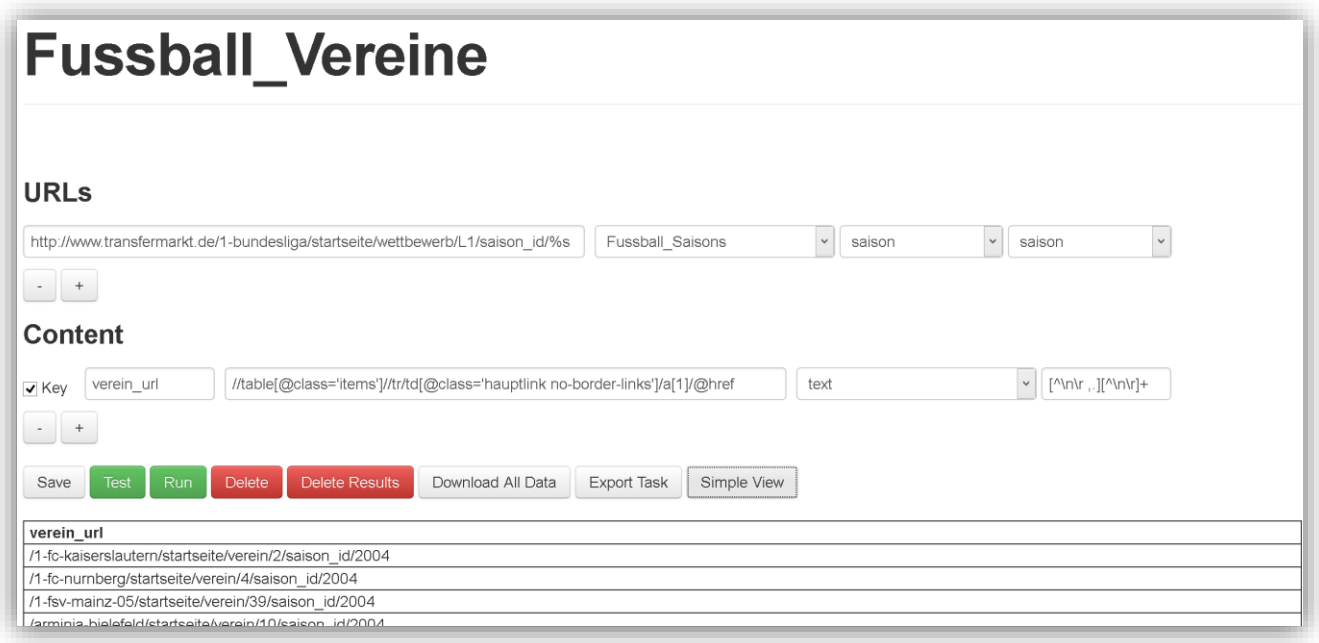

Abb. 3 Erweiterte Detailansicht einer Web-Scraping Aufgabe

URL-Selektoren können dynamisch auf Basis bereits gesammelter Daten spezifiziert werden. In diesem Fall wird die URL der Fußballvereine mit einem Platzhalter *%s* ausgestattet, der mit der Spalte *saison* der Tabelle *Fussball_Saisons* gefüllt wird.

Daten-Selektoren können außer den Rohdaten (Datentyp *string*) auch in andere Formate gewandelt werden. Bei Auswahl des Datentyps *float* wird beispielsweise automatisch der Reguläre Ausdruck `(\d[\d.,:]*)` angewendet und eine automatische Konvertierung des *strings* nach *float* durchge-

[28] https://addons.mozilla.org/de/firefox/addon/xpath-checker/ für Firefox beziehungsweise https://chrome.google.com/webstore/detail/xpath-helper/hgimnogjllphhhkhlmebbmlgjoejdpjl für Chrome

führt. Hierbei wird durch Kombination verschiedener Heuristiken Rücksicht auf international variierende Formate genommen. Gleiches gilt für Datumsangaben. Die Regulären Ausdrücke im letzten Eingabefeld lassen sich bei Bedarf manuell anpassen.

4.2.2. Exportieren der Daten

Die gesammelten Daten werden in einer Vorschau am unteren Ende der Seite angezeigt. Sie lassen sich mit proprietären Werkzeugen des verwendeten Datenbanksystems abfragen und exportieren. Zudem existiert eine Exportfunktion nach Excel. Falls sehr große Datenmengen nach Excel exportiert werden sollen, kann das HTTP-Timeout des Browsers heraufgesetzt werden. In Firefox befindet sich diese Einstellung auf der Seite `about:config` in `htttp.response.timeout`.

4.2.3. Testen der Aufgabenspezifikation

Mittels des Test-Buttons lässt sich eine Vorschau auf die gesammelten Daten anfordern, bei der nur die erste URL verwendet und keine Daten persistiert werden. Dies ist besonders bei der Erstellung neuer Aufgaben hilfreich.

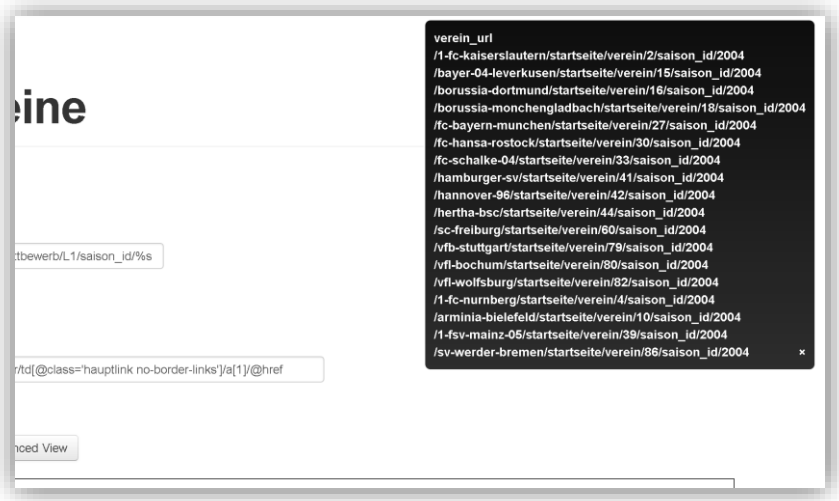

Abb. 4 Vorschau auf Web-Scraping Ergebnisse mit Hilfe des Testmodus

4.3. Anfragekonsole

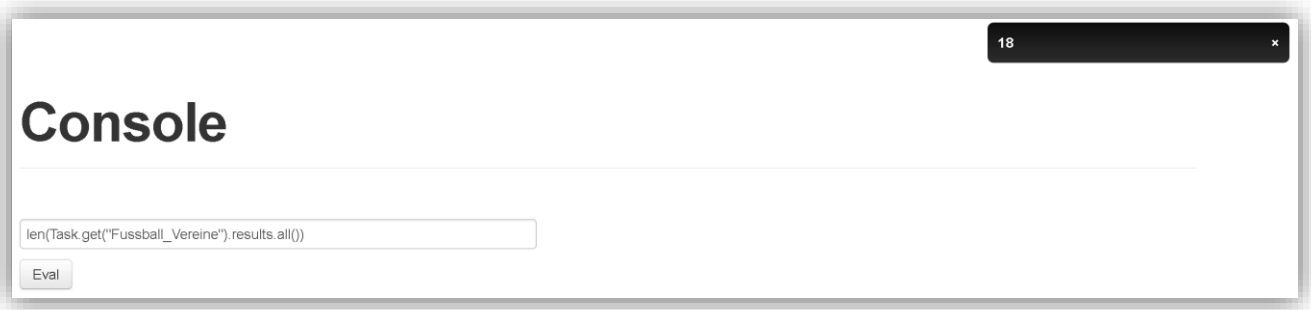

Abb. 5 Die Anfragekonsole

Im laufenden Betrieb des Servers kann es hilfreich sein, die Daten und Modelle programmiertechnisch abzufragen. Zu diesem Zwecke existiert eine Abfragekonsole unter der URL:
http://127.0.0.1:8080/idpscraper/console
Diese Möglichkeit der Abfrage ist gleichzeitig sehr mächtig, da über Python beliebige Verarbeitungen durchgeführt werden können. Sie muss aber im laufenden Betrieb unbedingt über eine Benutzerkontenregelung gesichert werden. Andernfalls kann beliebiger Programmcode auf dem Server ausgeführt werden. Alternativ lässt sich die Konsole auch deaktivieren.

4.4. Benutzer- und Rechteverwaltung

Da die *skalierbare* Version für externe Server gedacht ist, ist hier unmittelbar eine Rechteverwaltung aktiv. So muss bei Erstbenutzung ein Benutzerkonto angelegt werden, das fortan die vollen Benutzerrechte besitzt. Alle Unterseiten und Schnittstellen lassen sich anschließend nur angemeldet benutzen. Danach sollte die Registrierung im Programmcode deaktiviert werden.[29]

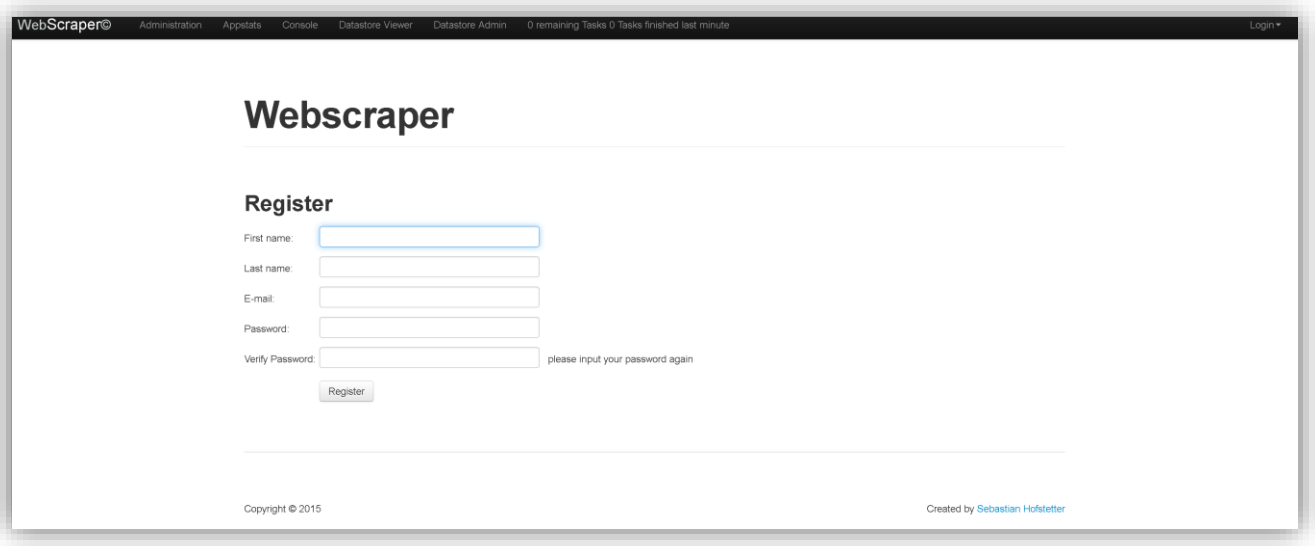

Abb. 6 Benutzerverwaltung des skalierbaren Systems

Die *effiziente* Version besitzt initial keine eingeschaltete Rechteverwaltung, da dies auf einem lokalen Server in der Regel nicht nötig ist. Sie kann aber einfach aktiviert werden.[30]

[29] http://web2py.com/books/default/chapter/29/09/access-control#Restrictions-on-registration
[30] https://docs.djangoproject.com/en/dev/topics/auth/?utm_medium=twitter&utm_source=twitterfeed

5. Anwendungen

5.1. Weltspitze der Leichtathletik

Die Auswertung der Leichtathletik-Weltspitze basiert auf den Daten der *International Association of Athletics Federation* http://www.iaaf.org . Es werden für die Jahre 1999 bis 2014 für jede Disziplin die jeweils 20 besten Athleten untersucht. Die Datenbasis umfasst 50758 Bestleistungen von insgesamt 16330 Athleten. Eine vollständige Liste der erhobenen Disziplinen findet sich im Anhang A2. Die Konfiguration des Web-Scrapers für alle folgenden Aufgaben können Anhang A1 entnommen werden.

5.1.1. Relative Age Effect

Der Relative Age Effect entsteht durch den Umstand, dass Athleten, die zu Beginn des Jahres geboren werden, im Vergleich zum übrigen Jahrgang weiter entwickelt sind. Die relative Abweichung hat besonders im Jugendalter Einfluss auf den Erfolg und auch die Förderung, die ein Athlet erfährt. Mit der Zeit nimmt die Verzerrung durch das Alter ab, die bereits erfahrene Mehrförderung bleibt aber in Form eines Erfahrungsvorsprungs erhalten.

Der Effekt lässt sich leicht nachweisen, indem man die Verteilung der Geburtsdaten untersucht. Bei Verwendung einer fremden Datenbasis muss darauf geachtet werden, dass das eingetragene Geburtstdatum unter Umständen nicht korrekt ist. Insbesondere wird oft der 1. Januar für Einträge verwendet, bei denen nur das Geburtsjahr bekannt ist. Aus diesem Grund werden im folgenden Athleten mit Geburtsdatum 1. Januar ignoriert, wodurch der Relative Age Effect nur unterschätzt aber nicht überschätzt werden kann. Von 16330 verbleiben 15385 verwertbare Geburtstage. Bereits im Sprung von Januar auf Februar zeigt der Chi-Quadrat-Test eine hochsignifikante Abweichung von einer Gleichverteilung.

Die Grundannahme der Gleichverteilung der Geburtsmonate lässt sich zwar unter Auswertung der Daten des statistischen Bundesamts 2013 nicht uneingeschränkt bestätigen. Jedoch liegt der Januar nahe dem Mittelwert, weshalb dieser zur Widerlegung der Nullhypthoese hergenommen werden darf.

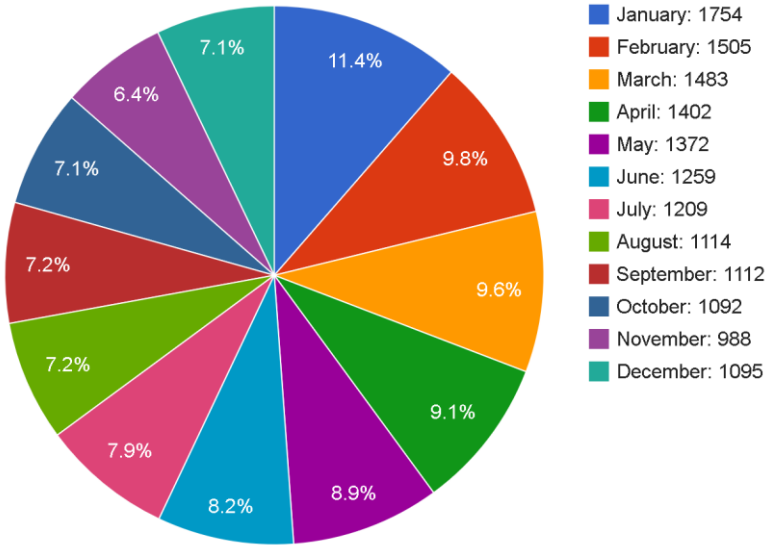

Abb. 7 Relative Age Effect der Leichtathletik Weltspitze von 1999-2014

5.1.2. Leistungsdaten

Nachdem die Leistungsdaten nach Excel exportiert sind, lassen sie sich leicht nach Nation, Altersklasse, Disziplin und weiteren Klassifikatoren sortieren und filtern. Da der Primäre Schlüssel eines Datensatzes aus dem Tupel (`athlete_id, datetime, class, disciplin`) besteht, können die Daten jedes Jahr leicht neu erhoben werden, ohne dass Duplikate entstehen.

rank	id	first name	last name	performance	date	gender	class	discpiplin	birthday	nation	area
1	176453	Justin	Gatlin	9.77	05.09.2014	men	senior	100-metres	10.02.1982	USA	outdoor
2	206487	Richard	Thompson	9.82	21.06.2014	men	senior	100-metres	07.06.1985	TTO	outdoor
3	189571	Asafa	Powell	9.87	23.08.2014	men	senior	100-metres	23.11.1982	JAM	outdoor
4	200897	Mike	Rodgers	9.91	20.07.2014	men	senior	100-metres	24.04.1985	USA	outdoor
5	185464	Tyson	Gay	9.93	03.07.2014	men	senior	100-metres	09.08.1982	USA	outdoor
5	253885	Femi	Ogunode	9.93	28.09.2014	men	senior	100-metres	15.05.1991	QAT	outdoor
5	262174	Kemarley	Brown	9.93	17.05.2014	men	senior	100-metres	20.07.1992	JAM	outdoor
8	248925	Jimmy	Vicaut	9.95	18.05.2014	men	senior	100-metres	27.02.1992	FRA	outdoor
9	200802	Nesta	Carter	9.96	21.08.2014	men	senior	100-metres	11.10.1985	JAM	outdoor
9	20646	Kim	Collins	9.96	20.07.2014	men	senior	100-metres	05.04.1976	SKN	outdoor
9	246675	Kemar	Bailey-Cole	9.96	28.08.2014	men	senior	100-metres	10.01.1992	JAM	outdoor
9	265959	Chijindu	Ujah	9.96	08.06.2014	men	senior	100-metres	05.03.1994	GBR	outdoor
13	227948	Nickel	Ashmeade	9.97	11.07.2014	men	senior	100-metres	07.04.1990	JAM	outdoor
13	273179	Trayvon	Bromell	9.97	13.06.2014	men	senior	100-metres	10.07.1995	USA	outdoor
15	184599	Usain	Bolt	9.98	23.08.2014	men	senior	100-metres	21.08.1986	JAM	outdoor
15	246210	Simon	Magakwe	9.98	12.04.2014	men	senior	100-metres	14.05.1986	RSA	outdoor
17	208570	Keston	Bledman	10.00	21.06.2014	men	senior	100-metres	08.03.1988	TTO	outdoor
17	231406	James	Dasaolu	10.00	05.09.2014	men	senior	100-metres	05.09.1987	GBR	outdoor
17	279646	Trentavis	Friday	10.00	05.07.2014	men	senior	100-metres	05.06.1995	USA	outdoor
20	238618	Antoine	Adams	10.01	22.06.2014	men	senior	100-metres	31.08.1988	SKN	outdoor

Abb. 8 Leistungsdaten der Leichtathletik Weltspitze seit 1999

5.2. Erste Deutsche Bundesliga im Fußball

Die Auswertung der Ersten Bundesliga basiert auf den Daten von http://www.transfermarkt.de/ . Untersucht werden die Verletzungshistorien, die in den Jahren 2008-2014 ausführlich vorliegen. Die Datenbasis umfasst 2063 Spieler.

5.2.1. Relative Age Effect

Der Relative Age Effect wird analog zu Kapitel 5.1.1 untersucht. Aufgrund der hohen Datenqualität können alle 2063 Geburtstage verwertet werden. Obwohl die relative Abweichung von Januar zu Februar nicht geringer als in der Leichtathletik ist, ist diese wegen der kleineren Datenbasis nicht signifikant. Vergleicht man jedoch den Januar mittels Chi-Quadrat-Test mit einem beliebigen Monat ab April, bestätigt sich der Effekt hochsignifikant.

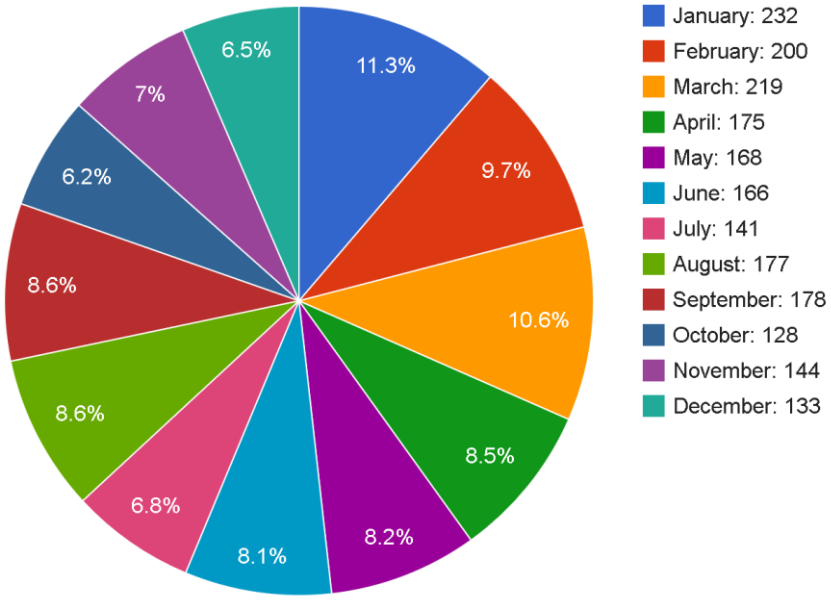

Abb. 9 Relative Age Effect der Fußballspieler der Ersten Deutschen Bundesliga von 2008-2014

5.2.2. Zusammenhang zwischen Spieleinsätzen und Verletzungen

Der Zusammenhang zwischen Spieleinsätzen und Verletzungen von Fußballspielern der Ersten Bundesliga wird aktuell am Lehrstuhl für Trainingswissenschaft und Sportinformatik der Technischen Universität München untersucht. Im Folgenden wird deshalb nur auf die Aggregation der Datenbasis, ohne Berücksichtigung der Auswertung, eingegangen.

Für die Untersuchung werden drei miteinander verknüpfte Tabellen erhoben, die um automatisch berechnete Spalten ergänzt werden. Die Liste der Spieler entsteht über mehrere Hilfsdatenbanken, die zunächst die in Frage kommenden Saisons, daraus die Vereine und schließlich die Kaderspieler ermittelt. Hier ist insbesondere das Datum des Karriereendes wichtig, da eine Beziehung zu einer schweren Verletzung hergestellt werden kann.

id	name	position	birthday	size	retire_date
10	Miroslav Klose	Mittelstürmer	09.06.1978	1.84	
100399	Nicolai Jörgensen	Hängende Spitze	15.01.1991	1.90	
101277	Márkó Futács	Mittelstürmer	22.02.1990	1.96	
10152	Cagdas Atan	Linker Verteidiger	29.02.1980	1.87	
102192	Luis Pedro Cavanda	Rechter Verteidiger	02.01.1991	1.80	
102226	Thorgan Hazard	Offensives Mittelfeld	29.03.1993	1.74	
1023	Godfried Aduobe	Defensives Mittelfeld	29.10.1975	1.78	01.07.2011
102382	Michael Hefele	Innenverteidiger	01.09.1990	1.92	
10248	Christian Müller	Offensives Mittelfeld	28.02.1984	1.82	
10254	Daniel Fernandes	Torwart	25.09.1983	1.95	
102558	Marco Verratti	Zentrales Mittelfeld	05.11.1992	1.65	
102740	Mário Fernandes	Rechter Verteidiger	19.09.1990	1.86	

Abb. 10 Spielerdaten der Fußballspieler der Ersten Deutschen Bundesliga

Aus der Spielerliste lassen sich alle Spieleinsätze aggregieren. Die Anzahl der gespielten Minuten dient als möglicher Indikator für die Exposition, unter der Verletzungen entstehen.

id	minutes_played	date	season
10	90	10.08.2008	2008
10	79	15.08.2008	2008
10	46	20.08.2008	2008
10	46	23.08.2008	2008
10	71	31.08.2008	2008
10	65	06.09.2008	2008
10	90	10.09.2008	2008
10	57	13.09.2008	2008
10	46	17.09.2008	2008
10		20.09.2008	2008
10	65	24.09.2008	2008
10	90	27.09.2008	2008
10	90	30.09.2008	2008
10	69	04.10.2008	2008

Abb. 11 Spieleinsätze der Ersten Deutschen Bundesliga Fußballspieler

Daraufhin werden die Verletzungen aller Spieler ermittelt. Durch eine Verknüpfung mit den Spieleinsätzen lässt sich mit hoher Wahrscheinlichkeit darauf schließen, wann eine Verletzung durch einen Spieleinsatz und wann durch ein Training verursacht wurde. Da die Daten nicht notwendigerweise exakt sind, wird eine Verletzung als *in_action* markiert, wenn eine sie in einem Zeitintervall der nächsten 48 Stunden eingetragen wurde. Die Dauer der Verletzung kann als Maß für die Verletzungsschwere herangezogen werden.

from	to	duration	missed_games	club	season	in_action	end date after present date?	preceding injury date	following injury date	preceeding injury in last year?	minutes played in season
27.02.2010	11.03.2010	12	2	Hannover 96	2009	0	FALSCH			FALSCH	
19.09.2008	22.09.2008	3	1	FC Bayern München	2008	0	FALSCH			FALSCH	
15.03.2009	05.05.2009	51	8	FC Bayern München	2008	1	FALSCH			FALSCH	
18.07.2009	22.07.2009	4			2009	0	FALSCH			FALSCH	
28.07.2009	01.08.2009	4			2009	0	FALSCH			FALSCH	
07.08.2009	11.08.2009	4	1	FC Bayern München	2009	0	FALSCH		07.02.2010	FALSCH	
25.11.2009	04.12.2009	9	3	FC Bayern München	2009	1	FALSCH			FALSCH	
07.02.2010	15.02.2010	8	2	FC Bayern München	2009	1	FALSCH	07.08.2009		WAHR	

Abb. 12 Verletzungen der Fußballspieler der Ersten Deutschen Bundesliga

Eine Aggregation über Spieler und Saison liefert schließlich eine Pivotabelle der Verletzungen.

id	season	Minutes Played	In Action Injury Count	Injury Count
6	2009	0	0	1
10	2008	3296	1	2
10	2009	2171	2	5
10	2010	2112	1	2
10	2011	90	0	0
26	2008	3345	1	1
26	2009	2901	1	1
26	2010	3900	0	1
26	2011	3874	0	1
26	2012	4401	1	2
26	2013	4125	3	3
26	2014	1394	0	1
29	2008	1500	0	1

Abb. 13 Verletzungen der Fußballspieler der Ersten Deutschen Bundesliga (aggregiert über die Spieler und die Saison)

Zuletzt kann eine Aggregation über die Monate indizien über die Verletzungsverteilung liefern.

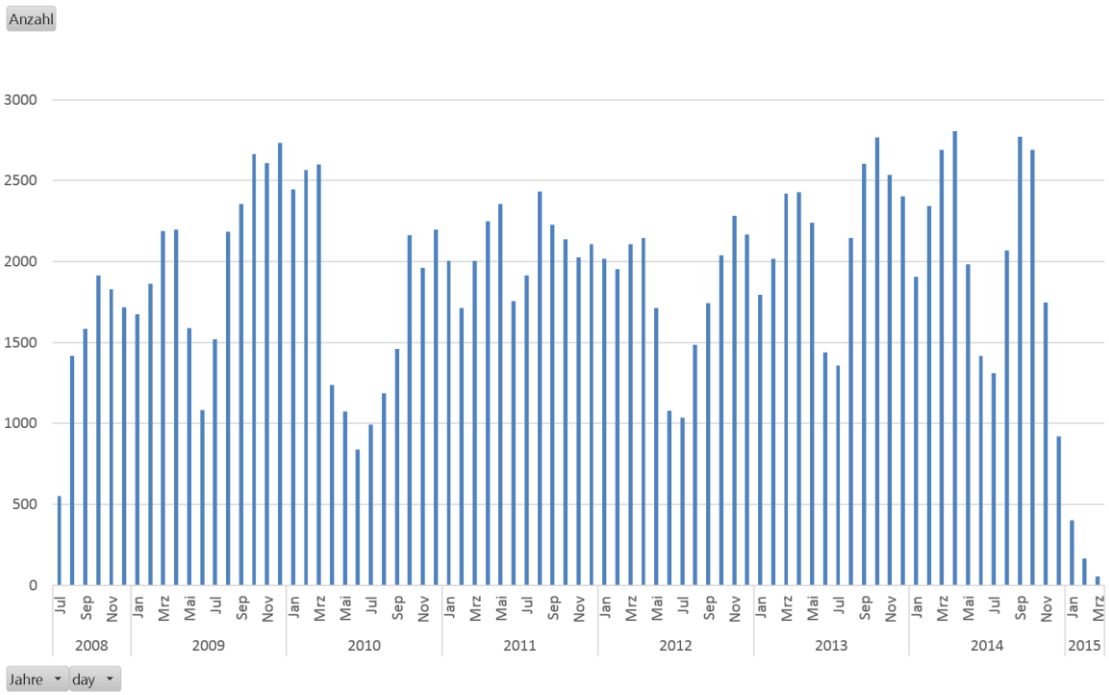

Abb. 14 Verletzungstage der Fußballspieler der Ersten Deutschen Bundesliga (aggregiert über den Monat)

5.2.3. Rückfallgefahr bei Verletzungen

Die Rückfallgefahr bei Verletzungen von Fußballspielern der Ersten Bundesliga wird aktuell am Lehrstuhl für Trainingswissenschaft und Sportinformatik der Technischen Universität München untersucht. Im Folgenden wird deshalb nur auf die Aggregation der Datenbasis, ohne Berücksichtigung der Auswertung, eingegangen.

Die Datenbasis entspricht der aus Abb. 12, Abb. 11 und Abb. 10. Um die Datenqualität zu verbessern, wird eine Synonymliste von ähnlichen Verletzungsbeschreibungen erstellt. Hieraus leiten sich die Spalten *preceding injury date* sowie *following injury date* ab. Aus diesen lässt sich wiederum eine Verletzungskette des jeweiligen Spielers berechnen. Die abschließenden Ergebnisse werden in Kürze vom Lehrstuhl veröffentlicht.

6. Zusammenfassung und Ausblick

Es wurde die Konzeption und Verwendung eines Web-Scraping Frameworks mit graphischer Oberfläche für den Einsatzbereich in der Trainingswissenschaft und Sportinformatik gezeigt. Alle Anforderungen an die Bedienung, Skalierbarkeit und Wiederbenutzbarkeit konnten erfüllt werden. Das Projekt steht auf GitHub als OpenSource Repository zur freien Verfügung und wird aktiv weiterentwickelt. Der Web-Scraper wird am Lehrstuhl für Trainingswissenschaft und Sportinformatik der Technischen Universität München für weitere Anwendungen der Forschung eingesetzt.

Die Leistungsdaten der Weltspitze der Leichtathletik können bereits jetzt für statistische Auswertungen herangezogen werden. Die Datenbasis der Verletzungshistorie der Ersten Bundesliga zur Untersuchung des Zusammenhangs zwischen Spieleinsätzen und Verletzungen sowie zur Untersuchung der Rückfallgefahr von Verletzungen ist fertiggestellt und wird zu diesem Zeitpunkt zu Forschungszwecken ausgewertet.

Mögliches Verbesserungspotenzial könnte zukünftig dadurch entstehen, dass freie NoSQL Datenbanksysteme die gleiche Leistung und transaktionale Sicherheit wie SQL Systeme liefern. Damit könnte die lineare Suche durch die Datenbank mittels Indexierung weiter beschleunigt werden.

Zusammenfassend lässt sich mit Hilfe dieses Projekts die Schwelle zur maschinellen Auswertung frei verfügbarer Daten senken und langfristig vielleicht sogar ganz aufheben.

II. Abbildungsverzeichnis

III. Anhang

A1. Konfigurationen der im Intedisziplinären Projekt erstellten Web-Scraping Aufgaben

```
Task(name="Fussball_Saisons"),
UrlSelector(task_id='Fussball_Saisons', url="http://www.transfermarkt.de/3262/kader/verein/3262/", selector_task_id='Fussball_Saisons', selector_name="saison", selector_name2="saison"),
Selector(task_id='Fussball_Saisons', name="saison", is_key=True, xpath='''//select[@name="saison_id"]/option/@value''', type=0, regex="2004"),

Task(name="Fussball_Vereine"),
UrlSelector(task_id='Fussball_Vereine', url="http://www.transfermarkt.de/1-bundesliga/startseite/wettbewerb/L1/saison_id/%s", selector_task_id='Fussball_Saisons', selector_name="saison", selec-
tor_name2="saison"),
Selector(task_id='Fussball_Vereine', name="verein_url", is_key=True, xpath='''//table[@class='items']//tr/td[@class='hauptlink no-border-links']/a[1]/@href''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name="Fussball_Spieler"),
UrlSelector(task_id='Fussball_Spieler', url="http://www.transfermarkt.de/%s", selector_task_id='Fussball_Vereine', selector_name="verein_url", selector_name2="verein_url"),
Selector(task_id='Fussball_Spieler', name="spieler_id", is_key=True, xpath='''//a[@class="spielprofil_tooltip"]/@href''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Spieler', name="saison", is_key=True, xpath='''//select[@name="saison_id"]/option[@selected="selected"]/@value''', type=0, regex="\\d[\\d.,]*"),

Task(name="Fussball_Einsaetze"),
UrlSelector(task_id='Fussball_Einsaetze', url="http://www.transfermarkt.de/spieler/leistungsdatendetails/spieler/%s/plus/1/saison/%s", selector_task_id='Fussball_Spieler', selector_name="spieler_id",
selector_name2="saison"),
Selector(task_id='Fussball_Einsaetze', name="spieler_id", is_key=True, xpath='''(//a[@class="megamenu"])[1]/@href''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Einsaetze', name="minutes_played", is_key=False, xpath='''//div[@class="responsive-table"]/table//tr/td[2]/following-sibling::*[last()]''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Einsaetze', name="date", is_key=True, xpath='''//div[@class="responsive-table"]/table//tr/td[2]''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name="Fussball_Spieler_Details"),
UrlSelector(task_id='Fussball_Spieler_Details', url="http://www.transfermarkt.de/daten/profil/spieler/%s", selector_task_id='Fussball_Spieler', selector_name="spieler_id", selec-
tor_name2="spieler_id"),
Selector(task_id='Fussball_Spieler_Details', name="spieler_id", is_key=True, xpath='''//link[@rel="canonical"]/@href''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Spieler_Details', name="name", is_key=False, xpath='''//div[@class="spielername-profil"]/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Spieler_Details', name="position", is_key=False, xpath='''//table[@class="profilheader"]//td[preceding-sibling::th/text()="Position:"]''', type=1, regex="[^\\n\\r
,.][^\\n\\r]+"),
Selector(task_id='Fussball_Spieler_Details', name="max_value", is_key=False, xpath='''//table[@class="auflistung mt10"]/tr[3]/td/text()''', type=3, regex="\\d[\\d.,:]*"),
Selector(task_id='Fussball_Spieler_Details', name="birthday", is_key=False, xpath='''//td[preceding-sibling::th/text()="Geburtsdatum:"]/a/text()''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Spieler_Details', name="size", is_key=False, xpath='''//td[preceding-sibling::th/text()="GrÃ¶ÃŸe:"]//text()''', type=3, regex="\\d[\\d.,:]*"),
Selector(task_id='Fussball_Spieler_Details', name="retire_date", is_key=False, xpath='''//table[@class="profilheader"]//td[preceding-sibling::*[.//@title="Karriereende"]]''', type=2, regex="[^\\n\\r
,.][^\\n\\r]+"),

Task(name="Fussball_Transfers"),
UrlSelector(task_id='Fussball_Transfers', url="http://www.transfermarkt.de/daten/profil/spieler/%s", selector_task_id='Fussball_Spieler', selector_name="spieler_id", selector_name2=""),
Selector(task_id='Fussball_Transfers', name="spieler_id", is_key=False, xpath='''(//a[@class="megamenu"])[1]/@href''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Transfers', name="date", is_key=False, xpath='''(//table)[3]//tr/td[2]//text()''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Transfers', name="from", is_key=False, xpath='''(//table)[3]//tr/td[5]/a/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Transfers', name="to", is_key=False, xpath='''(//table)[3]//tr/td[8]/a/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Transfers', name="transfer_key", is_key=True, xpath='''merge_lists((//a[@class="megamenu"])[1]/@href, (//table)[3]//tr/td[5]/a/text(), (//table)[3]//tr/td[8]/a/text())''',
type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name="Fussball_Verletzungen"),
UrlSelector(task_id='Fussball_Verletzungen', url="http://www.transfermarkt.de/spieler/verletzungen/spieler/%s", selector_task_id='Fussball_Spieler', selector_name="spieler_id", selec-
tor_name2="spieler_id"),
UrlSelector(task_id='Fussball_Verletzungen', url="http://www.transfermarkt.de%s", selector_task_id='Fussball_Verletzungen', selector_name="next_page", selector_name2="spieler_id"),
Selector(task_id='Fussball_Verletzungen', name="spieler_id", is_key=True, xpath='''(//a[@class="megamenu"])[1]/@href''', type=0, regex="\\d[\\d.,]*"),
```

```
Selector(task_id='Fussball_Verletzungen', name="injury", is_key=False, xpath='''//table[@class="items"]//tr/td[2]/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Verletzungen', name="from", is_key=True, xpath='''//table[@class="items"]//tr/td[3]/text()''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Verletzungen', name="to", is_key=False, xpath='''//table[@class="items"]//tr/td[4]/text()''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Verletzungen', name="duration", is_key=False, xpath='''//table[@class="items"]//tr/td[5]/text()''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Verletzungen', name="missed_games", is_key=False, xpath='''//table[@class="items"]//tr/td[6]/text()''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Fussball_Verletzungen', name="next_page", is_key=False, xpath='''//li[@class="naechste-seite"]/a/@href''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Fussball_Verletzungen', name="club", is_key=False, xpath='''exe(//table[@class="items"]//tr/td[6],".//@title")''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name='Leichtathletik_Saisons'),
Selector(task_id='Leichtathletik_Saisons', name='saison', type=0, xpath="id('selectyear')/option/@value", regex='\\d\\d\\d\\d', is_key=True),
UrlSelector(task_id='Leichtathletik_Saisons', url='http://www.iaaf.org/results', selector_task_id='Leichtathletik_Saisons', selector_name='', selector_name2=''),

Task(name="Leichtathletik_Disziplinen"),
UrlSelector(task_id='Leichtathletik_Disziplinen', url="http://www.iaaf.org/athletes", selector_task_id='Leichtathletik_Disziplinen', selector_name="disciplin", selector_name2=""),
Selector(task_id='Leichtathletik_Disziplinen', name="disciplin", is_key=True, xpath='''//select[@id="selectDiscipline"]/option/@value''', type=1, regex=""),

Task(name="Leichtathletik_Athleten"),
UrlSelector(task_id='Leichtathletik_Athleten', url="http://www.iaaf.org/athletes/search?name=&country=&discipline=%s&gender=", selector_task_id='Leichtathletik_Disziplinen', selector_name="disciplin",
selector_name2=""),
Selector(task_id='Leichtathletik_Athleten', name="athlete_id", is_key=True, xpath='''//table[@class="records-table"]//tr[not(@class)]/td[1]//@href''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Leichtathletik_Athleten', name="first_name", is_key=False, xpath='''//table[@class="records-table"]//tr[not(@class)]/td[1]//a/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Athleten', name="last_name", is_key=False, xpath='''//table[@class="records-table"]//tr[not(@class)]/td[1]/a/span/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Athleten', name="sex", is_key=False, xpath='''//table[@class="records-table"]//tr[not(@class)]/td[2]/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Athleten', name="country", is_key=False, xpath='''//table[@class="records-table"]//tr[not(@class)]/td[3]/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Athleten', name="birthday", is_key=False, xpath='''//table[@class="records-table"]//tr[not(@class)]/td[4]/text()''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name="Leichtathletik_Performance"),
UrlSelector(task_id='Leichtathletik_Performance', url="http://www.iaaf.org/athletes/athlete=%s", selector_task_id='Leichtathletik_Athleten', selector_name="athlete_id", selector_name2=""),
Selector(task_id='Leichtathletik_Performance', name="athlete_id", is_key=False, xpath='''//meta[@name="url"]/@content''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Leichtathletik_Performance', name="performance", is_key=False, xpath='''//div[@id="panel-progression"]//tr[count(td)>3]//td[2]''', type=3, regex="\\d[\\d.,:]*"),
Selector(task_id='Leichtathletik_Performance', name="datetime", is_key=False, xpath='''merge_lists(//div[@id="panel-progression"]//tr[count(td)>3]/td[last()], //div[@id="panel-progres-
sion"]//tr[count(td)>3]/td[1])''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Performance', name="place", is_key=False, xpath='''//div[@id="panel-progression"]//tr[count(td)>3]//td[last()-1]''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Performance', name="discipline", is_key=False, xpath='''exe(//div[@id="panel-progression"]//tr[count(td)>3]//td[2], "../preceding::tr/td[@class='sub-title']")''',
type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Performance', name="performance_key", is_key=True, xpath='''merge_lists(//div[@id="panel-progression"]//tr[count(td)>3]/td[last()], //div[@id="panel-progres-
sion"]//tr[count(td)>3]/td[1], //meta[@name="url"]/@content)''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name="Leichtathletik_Sprint_100m_Herren"),
UrlSelector(task_id='Leichtathletik_Sprint_100m_Herren', url="http://www.iaaf.org/records/toplists/sprints/100-metres/outdoor/men/senior", selector_task_id='Leichtathletik_Sprint_100m_Herren', selec-
tor_name="athlete_id", selector_name2=""),
Selector(task_id='Leichtathletik_Sprint_100m_Herren', name="athlete_id", is_key=True, xpath='''//table[@class = "records-table toggled-table condensedTbl"]/tr[@id]/td[4]/a/@href''', type=0, re-
gex="\\d[\\d.,]*"),
Selector(task_id='Leichtathletik_Sprint_100m_Herren', name="first_name", is_key=False, xpath='''//table[@class = "records-table toggled-table condensedTbl"]/tr[@id]/td[4]/a/text()''', type=1, re-
gex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Sprint_100m_Herren', name="last_name", is_key=False, xpath='''//table[@class = "records-table toggled-table condensedTbl"]/tr[@id]/td[4]/a/span/text()''', type=1,
regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Sprint_100m_Herren', name="result_time", is_key=False, xpath='''//table[@class = "records-table toggled-table condensedTbl"]/tr[@id]/td[2]/text()''', type=3, re-
gex="\\d[\\d.,:]*"),
Selector(task_id='Leichtathletik_Sprint_100m_Herren', name="competition_date", is_key=False, xpath='''//table[@class = "records-table toggled-table condensedTbl"]/tr[@id]/td[9]/text()''', type=2,
regex="[^\\n\\r ,.][^\\n\\r]+"),

Task(name="Leichtathletik_Top_Urls"),
UrlSelector(task_id='Leichtathletik_Top_Urls', url="http://www.iaaf.org/records/toplists/sprints/100-metres/outdoor/men/senior", selector_task_id='Leichtathletik_Top_Urls', selector_name="", selec-
tor_name2=""),
Selector(task_id='Leichtathletik_Top_Urls', name="url", is_key=True, xpath='''//input[@type="radio"]/@value''', type=1, regex=""),

Task(name="Leichtathletik_Top_Performance"),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/1999", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
```

```
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2000", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2001", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2002", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2003", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2004", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2005", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2006", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2007", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2008", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2009", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2010", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2011", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2012", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2013", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
UrlSelector(task_id='Leichtathletik_Top_Performance', url="http://www.iaaf.org%s/2014", selector_task_id='Leichtathletik_Top_Urls', selector_name="url", selector_name2=""),
Selector(task_id='Leichtathletik_Top_Performance', name="athlete_id", is_key=True, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]//@href''', type=0, regex="\\d[\\d.,]*"),
Selector(task_id='Leichtathletik_Top_Performance', name="first_name", is_key=False, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td/a/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="last_name", is_key=False, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td/a/span/text()''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="performance", is_key=False, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td[2]/text()''', type=3, regex="\\d[\\d.,:]*"),
Selector(task_id='Leichtathletik_Top_Performance', name="datetime", is_key=True, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td[last()]/text()''', type=2, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="gender", is_key=False, xpath='''//meta[@property="og:url"]/@content''', type=1, regex=".+/([^/]+)/[^/]+/[^/]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="class", is_key=True, xpath='''//meta[@property="og:url"]/@content''', type=1, regex=".+/([^/]+)/[^/]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="discpiplin", is_key=True, xpath='''//meta[@property="og:url"]/@content''', type=1, regex=".+/([^/]+)/[^/]+/[^/]+/[^/]+/[^/]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="birthday", is_key=False, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td[preceding-sibling::td[position()=1 and ./a]]''', type=2, re-
gex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="nation", is_key=False, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td/img/@alt''', type=1, regex="[^\\n\\r ,.][^\\n\\r]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="area", is_key=False, xpath='''//meta[@property="og:url"]/@content''', type=1, regex=".+/([^/]+)/[^/]+/[^/]+/[^/]+"),
Selector(task_id='Leichtathletik_Top_Performance', name="rank", is_key=False, xpath='''(//table)[1]//tr[.//a and ./td[1] <= 20]/td[1]''', type=0, regex="\\d[\\d.,]*"),
```

A2. Liste der erhobenen Leichtathletik Disziplinen mit Anzahl an erfassten Leistungen

Disziplin	Anzahl an Leistungen
30000-metres	3
30000-metres-race-walk	4
25000-metres	6
two-miles	8
50000-metres-race-walk	12
20000-metres	14
one-hour	17
decathlon-boys	20
hammer-throw-3kg	60
heptathlon-girls	60
shot-put-3kg	60
javelin-throw-500g	60
heptathlon-100mh-762cm	155
110m-hurdles-990cm	173
octathlon-boys	196
2000-metres	201
discus-throw-1750kg	227
shot-put-6kg	229
decathlon-junior	230
hammer-throw-6kg	231
3000-metres-race-walk	233
hammer-throw-5kg	240
javelin-throw-700g	241
400m-hurdles-840cm	241
discus-throw-1500kg	241
shot-put-5kg	242
100m-hurdles-762cm	246
110m-hurdles-914cm	247
50-metres-hurdles	262
50-kilometres-race-walk	301
pentathlon	322
50-metres	344
decathlon	347
20000-metres-race-walk	351
110-metres-hurdles	421
2000-metres-steeplechase	437
25-kilometres	493
100-kilometres	501
30-kilometres	505
20-kilometres	508
10-kilometres-race-walk	515
10-kilometres	521
5000-metres-race-walk	554
15-kilometres	554
100-metres-hurdles	564
marathon	601
20-kilometres-race-walk	604
half-marathon	606
60-metres	652
60-metres-hurdles	673
heptathlon	831
10000-metres-race-walk	843
hammer-throw	1011
10000-metres	1058
discus-throw	1093
3000-metres-steeplechase	1116
1000-metres	1176
javelin-throw	1242
one-mile	1282
400-metres-hurdles	1360
100-metres	1615
5000-metres	1622
shot-put	1658
triple-jump	2166
3000-metres	2174
800-metres	2201
1500-metres	2201
400-metres	2203
200-metres	2229
long-jump	2236
pole-vault	2426
high-jump	2482
Gesamtergebnis	**50758**