Olympia und Wikipedia

Olympia und Wikipedia

Analyse eines Korpus der Artikel zur Olympiade 2012

Bronder Benjamin, Homburg Timo

2. Februar 2019

Rüdiger Gleim

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument dient der Dokumentation der Aufgabe " Olympia in der Wikipedia " Auf den folgenden Seiten erklären wir zunächst was sich hinter dem Titel verbirgt, welches Resultat wir erzielen wollen und warum. Danach werden wir sowohl abstrakt das Vorgehen beschreiben, das zu unserem Resultat führt, als auch einen Einblick in den Lösungsfindungsprozess geben. Anschließend werden wir unsere Implementierung im Detail beschreiben. Abschließend wollen wir die Resultate analysieren, das heißt eine

Prüfung unseres Resultats auf seine Richtigkeit sowie in der Frage ob Resultat und Anfangsannahme zusammenpassen durchführen.

1.1 Problemstellung

In der Wikipedia finden sich viele verschiedene Artikel zum Thema Olympische Spiele 2012. Diese wurden von verschiedenen Benutzern, zu verschiedenen Zeitpunkten erstellt, sowie editiert und in unterschiedliche Kategorien eingeordnet. Die Darstellung der Revisionen in diesem Zeitraum zu Zwecken der Statistik, sowie die systematische Ausgabe der Attribute der betroffenen Artikel für weitere Analysen in einem automatisierten Prozess ist Aufgabe dieses Projektes. Weiterhin sollen Zusammenhänge zwischen bestimmten Artikeln als RDF Graph modelliert werden, um Sportler, Nationen und Sportarten in Beziehung zu setzen. Aufbauend auf den Beziehungen der Artikel können Rückschlüsse auf das Editierverhalten gezogen werden.

1.2 Motivation

Durch die Anwendung wird es möglich sein automatisiert auf die Artikel der Olympischen Spiele der Wikipedia zuzugreifen und die Korpusinformationen in einem vordefinierten Format zu extrahieren. Diese können für weitere Tests verwendet werden. Desweiteren stehen die Rohdaten für Statistiken der Revisionen der Artikel der Kategorie zur Verfügung um konkrete Ereignisse während der Olympischen Spiele in dem Datenverlauf anzuordnen. Durch einen Wechsel der URL und einiger weiterer Parameter, kann das Programm auch für andere Kategorisierungen verwendet werden und ist nicht nur auf die Kategorie Olympische Spiele begrenzt. Zuletzt kann eine Einordnung der verschiedenen Artikel im Korpus aufgrund gegebener Parameter vorgenommen werden. Im Rahmen dieser Ausarbeitung werden die Metadaten sowie Tabellen und Textinhalte der Artikel im Korpus herangezogen um um das Netzwerk der Beziehungen zwischen Sportlern, Sportarten ,Nationen und Medaillen zu erzeugen. Da dies in Form einer RDF-Datei geschieht ist das Resultat auch für andere Applikationen weiter zu nutzen.

2 Lösungsweg

2.1 Ausgangslage

Als Ausgangslage des Projektes fanden wir die API der Wikipedia vor. Diese bot uns verschiedene Anfragearten, die uns bei der Programmierung von Nutzen waren. Weiterhin wurde die Möglichkeit der Benutzung der Wikipedia XML Dumps für ergänzende Informationen zur Verfügung gestellt. Grafiken, die zur Visualisierung der Ergebnisse dienen, sollten mithilfe von Matlab erstellt werden. Der Export der Ergebnisse sollte, die Zusammenhangsgraphen betreffend, im Resource Description Format (RDF) erfolgen. Weitere Exports sollten als durch Tabs separierte Textdateien erfolgen. Die Anforderungen an das Programm beinhalteten die automatische Beschaffung der notwendigen Daten, sowie die bereits beschriebene geforderte Funktionalität.

2.2 Vorgehensweise

Begonnen haben wir mit der vorgeschlagenen Kategorie Olympische Sommerspiele 2012 als Hauptbezugspunkt. Um den Umfang des Korpus auf den direkten bis mittelbaren Kontext zu beschränken wählten wir die Artikel, die direkt zu der Kategorie gehören sowie jene die zu einer Subkategorie gehören und auf die Hauptkategorie zurückzuführen sind. Mithilfe der Wikipedia API erhielten wir so die Liste der gewünschten Titel, die wir rekursiv abarbeiten konnten um den Korpus aus "Seitenobjekten" zu erstellen, die die Metadaten enthalten. Herausstechende Peaks im resultierenden Plot haben wir zusätzlich auf zeitgleiche Ereignisse überprüft die mit der Olympiade in Verbindung stehen.

Da der interessante Punkt dieser Untersuchung die Aktivität bezüglich des Hauptthemas ist, und nicht die der einzigen Seiten, bot es sich an eine Map anzulegen, die für die Tage beim Einlesen der Revisionensdaten mitzählt. Diese Map enthielt ursprünglich die genauen Datumsangaben. Da sich diese jedoch nicht ohne weiteres in einer gut lesbaren Form in Matlab plotten ließen stiegen wir auf "Tag x nach ersten Wikipedia Eintrag im Corpus" um.

Zum Schluss sollten noch Sportler, Sportarten und Nationen aus dem Korpus gefiltert, ihre Verknüpfungen rekonstruiert und als Zusammenhangsgraph im RDF Format gespeichert werden. Dabei stellte sich der mangelnde Umfang der Artikel in der deutschsprachigen Wikipedia als größtes Hindernis heraus. Der überwiegende Teil der Sportler, die nicht für Deutschland teilnahmen, haben keine eigene Seite in der deutschsprachigen Wikipedia. Sie wären folglich nur als Dummy-links in anderen Seiten zu finden, was ein sicheres zuweisen als Sportler (oder gar Person) unmöglich gemacht hätte. Aus diesem Grund sind wir auf die englischsprachige Wikipedia umgestiegen.

2.2.1 Entitäten

Die Nationen lassen sich aus der Liste der teilnehmenden Nationen ermitteln. Die Sportarten aus der Liste der Disziplinen/Events. Da die einzige gemeinsame Kategorie aller Sportler "Category:Living People" ist erstellten wir eine Vorsortierung potentieller Teilnehmer in welcher wir die ungewollten Einträge (z.B.Mick Jagger) dadurch abfangen, das wir überprüfen ob eine ihrer Kategorien " at 2012 Summer Olympics" enthält.

Ein Fehler der dabei aufgefallen ist, sich aber nicht automatisch verhindern lässt, sind nicht einheitliche Wikipedia Kategorien. Beispielsweise wird Beachvolleyball offiziell als Unterart des Volleyballs geführt und ist somit in der Eventliste keine eigenständige Kategorie. Allerdings hat keiner der Beachvolleyball-Spieler "Volleyball" in irgendeiner Form als zugeordnete Kategorie. Da wir das Programm aber möglichst robust und einfach übertragbar halten wollten, entschieden wir uns dagegen Fehler manuell abzufangen, die auch falsch in der Wikipedia stehen. Diese Listen werden zur Veranschaulichung auch separat ausgegeben.

2.2.2 Relationen

Wenn sich alle Seiten an die Vorlage halten, enthält jede Seite zu einem potentiellen Sportler eine relative kurze Liste von Links von denen einer zu seiner Nation führt. Als ersten Ansatz um doppelte Staatsbürgerschaften und Auswanderer abzufangen prüften wir ob der Sportler einen Link zu einer Nation hat noch ob sie einen Backlink von einer "Olympiateilnehmer von (Nation) 2012" Liste enthalten. Da leider ein signifikanter Teil der Sportlerseiten keinen Link zu einer Nation enthält, prüften wir stattdessen "nur" ob es einen Backlink von einer " Nation at the 2012 Summer Olympics" Seite gibt wenn der erste Vergleich kein Ergebnis liefert.

Um die Anzahl der Sportler deren Nation "Unbekannt" ist zu reduzieren gehen folglich das Risiko ein mehr false positivs zu erhalten. Denn auch wenn es unwahrscheinlich ist, so ist es nicht ganz auszuschließen das ein Sportler von von einer " Nation at the 2012 Summer Olympics" Page verlinkt wird für die er nicht angetreten ist.

Die Sportler hingegen sind (bis auf die Beach-Volleyballer) ordentlich Katalogisiert. So sind z.B. alle teilnehmenden Fechter unter "Fencers at the 2012 Summer Olympics" zu finden, was eine gewisse Ähnlichkeit zur Veranstaltung "Fencing at the 2012 Summer Olympics" aufweist. Folglich lässt sich das Relationstripel "Sportler betreibt/tritt-an Sport" einfach ermitteln.

Wenn wir wissen, das ein Sportler für eine Nation teilgenommen hat und in welcher Sportart, dann lässt sich zwischen Nation und Sportart auch eine direkte Kante eintragen.

Die Medaillenspiegel sind leider nicht aus den Metadaten zu entnehmen und bei weitem nicht einheitlich aufgebaut. Um sie zu extrahieren waren einige Extraschritte notwendig die, da in der Wikipedia nicht mit wohlgeformten Content zu rechnen ist, wir lediglich für die Sportarten/Events Seiten anwenden.

Zu diesem Zweck haben wir einen DefaultHTTP Client erstellt und mit diesem einen HTTPGet Request auf den Seiten URLs ausgeführt. Danach haben wir die Ersetzungen auf dem HTML Text durchgeführt um es XMLKonformität zu erreichen und anschließend mit einem StringReader über einen InputStream die Infos dem SAXParser übergeben, um mit diesem die Medailleninformationen zu parsen.

Aus der Liste der Sportler und der Information das jemand mit dem Namen Gold/Silber/Bronze gewonnen hat, lassen sich die Sportler/Medaillen Relationen anlegen. Aus dieser Relation und der Relation zwischen Sportler und Relation lässt sich schließlich auch die Verbindung zwischen Nationen und Medaillen knüpfen.

2.2.3 RDF Export

Abschließend haben wir all diese Relationen genommen und als RDF-XML gespeichert.

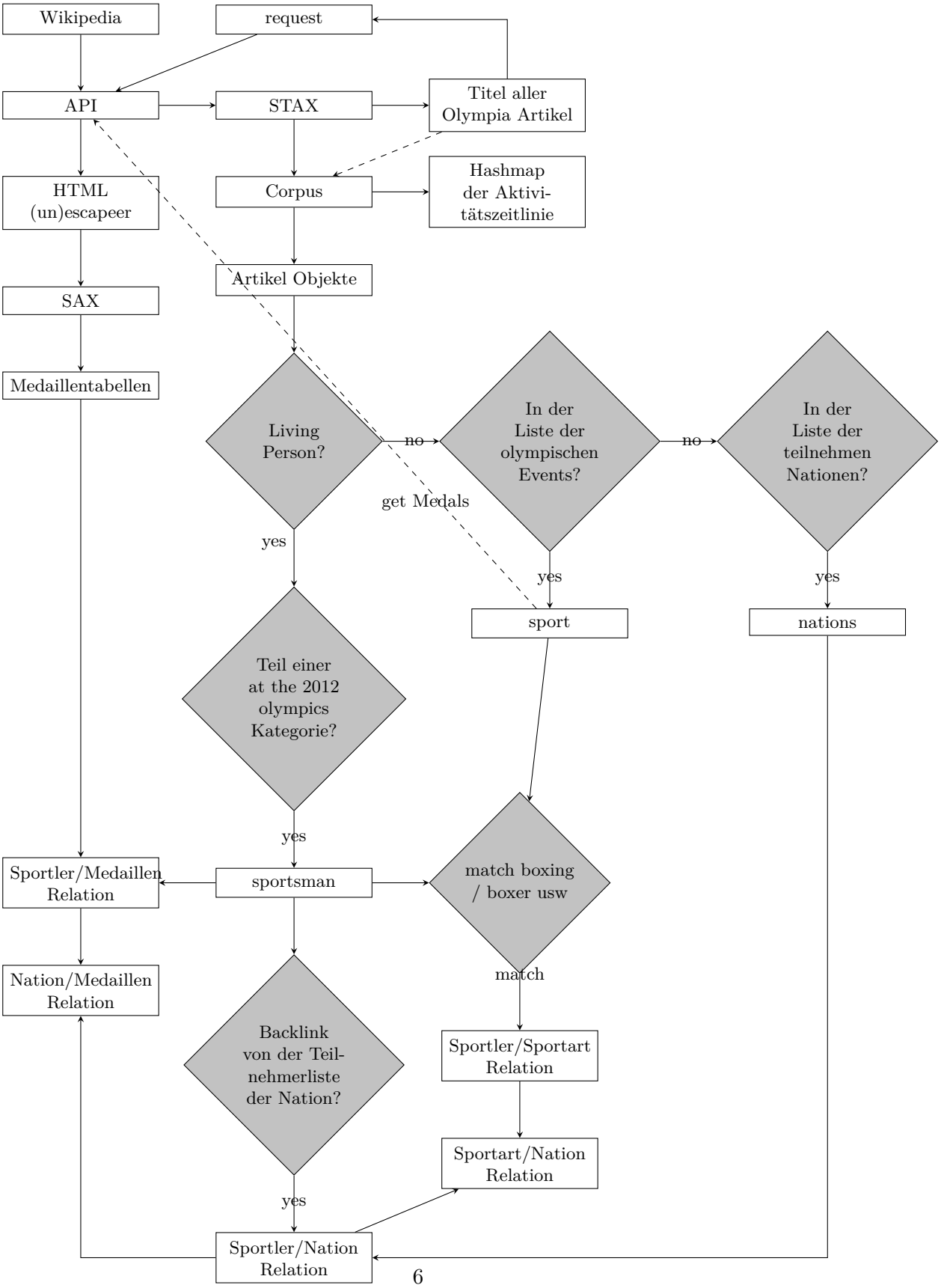

Abbildung 1: Flussdiagramm des Lösungsansatzes

3 Implementierung

3.1 Architektur

Die Architektur des Programms lässt sich in verschiedene Funktionseinheiten aufteilen:

- Ermitteln der relevanten Kategorien
- Erstellen des Korpus der relevanten Artikel und Ermittlung der Entitäten
- Export des Korpus
- Analyse der Revisionsinformationen und Export der relevanten Revisionsdaten
- Herstellung von Beziehungen zwischen den Entitäten
- Export der Beziehungen im RDF-Format

Das folgende Klassendiagramm soll eine Übersicht über die Beziehungen der nachfolgend beschriebenen Klassen bieten:

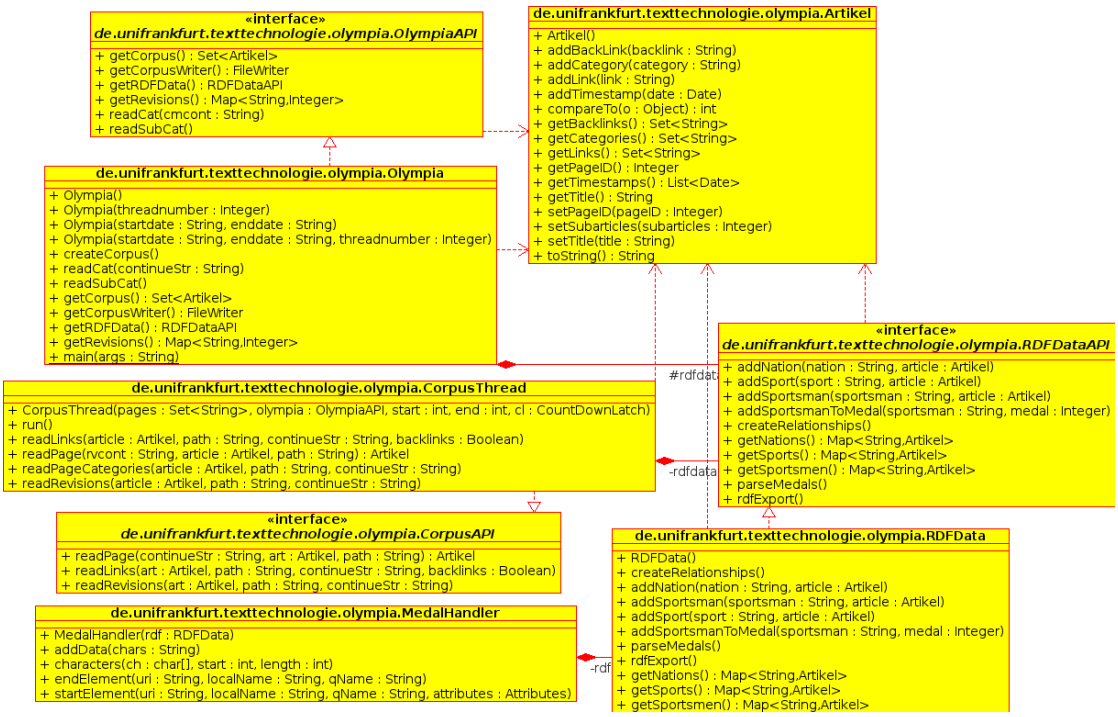

Abbildung 2: Klassendiagramm

3.1.1 Erstellung des Korpus

Für das Ermitteln der für den Korpus relevanten Kategorien wurde von der Basiskategorie ”2012 Summer Olympics” ausgehend eine Analyse der Unterkategorien durchgeführt. Durch eine rekursive Weiterführung dieses Prozesses wurden sämtliche Kategorien die in irgendeiner Form mit den Olympischen Spielen in Verbindung standen extrahiert. Die genutzten URLs der Wikipedia API und weitere häufig verwendete Strings wurden in den zugehörigen Interfaces zur einfachen Konfiguration und Wiederverwendung zwischengespeichert. Beispiele für verwendete API Aufrufe finden sich hier:

```
http://en.wikipedia.org/w/api.php?action=query&list=categorymembers
&cmtitle=Category:2012_Summer_Olympics&format=xml
<!-- Kategorie API Aufruf -->
http://en.wikipedia.org/w/api.php?action=query
&titles=American_Samoa_at_the_2012_Summer_Olympics&format=xml
<!-- Page Parse API Aufruf -->
http://en.wikipedia.org/w/api.php?action=query
&titles=American_Samoa_at_the_2012_Summer_Olympics
&prop=categories&format=xml
<!-- Kategorien der Seite -->
http://en.wikipedia.org/w/api.php?action=query
&titles=American_Samoa_at_the_2012_Summer_Olympics
#&prop=links&format=xml
<!-- Links der Seite -->
http://en.wikipedia.org/w/api.php?action=query
&titles=American_Samoa_at_the_2012_Summer_Olympics
&bllimit=max&list=backlinks
&bltitle=American_Samoa_at_the_2012_Summer_Olympics&format=xml
<!-- Backlinks der Seite -->
http://en.wikipedia.org/w/api.php?action=query
&titles=American_Samoa_at_the_2012_Summer_Olympics
&rvlimit=max&prop=revisions&rvprop=timestamp|user
<!-- Revisionen der Seite -->
http://en.wikipedia.org/w/api.php?action=query
&titles=American_Samoa_at_the_2012_Summer_Olympics
&format=xml&rvcontinue=.....
<!-- Continue Strings (Bsp. fuer Revisionen, auch fuer Backlinks, Links und Kategorien benutzt)-->
```

Für die in der letzten URL erwähnten Continuestrings war es nötig je eine Methode rekursiv aufzurufen um diese abzuarbeiten. Ein Beispiel bezüglich des Auslesens von Kategorien findet sich hier:

```
        String cmcontinue = "";
        //Lese Kategorieinformationen u.a. auch den Rekursionsparameter aus
        .....
        //Fuehre die Rekursion aus um weitere Kategorien zu lesen
                if (!"".equals(cmcontinue))
                        this.readCat(cmcontinue);
}
```

Ausgehend von den Kategorien konnte als nächstes der Korpus erstellt werden. Für die Speicherung des Korpus wurde ein Objekt namens "Artikel" erstellt:

```
    /**Unique page id of the Wikipedia article.*/
  private Integer pageID;
  /**List of timestamps of the revisions of the article.*/
  private final List<Date> timestamps;
  /**List of categories of the article.*/
  private final Set<String> categories;
  /**List of links included in the article.*/
  private final Set<String> links;
  /**List of links included in the article.*/
  private final Set<String> backlinks;
  /**Amount of subarticles of this article.*/
  private transient Integer subarticles;
  /**Title of the article.*/
  private String title;
}
```

Dieses enthielt neben dem Titel der Seite, den Kategorien und der PageID auch die Links und Backlinks der jeweiligen Seite, die zur Herstellung von Beziehungen der Entitäten hilfreich sind. Die Timestamps der Revisionen des jeweiligen Artikels wurden bei dieser Gelegenheit ebenso aufgenommen.
Die Erstellung der Artikel wurde aufgrund des sehr langen Rechenprozesses sowie des sehr häufigen Internetzugriffs in mehrere Threads ausgelagert. Die Klasse CorpusThread repräsentiert diesen Thread und wird über die createCorpus() Methode in der Klasse Olympia aufgerufen. Die Funktionialität von CorpusThread teilt sich unterdessen in Funktionen zum Lesen von Revisionen, Links, Backlinks, Kategorien und den allgemeinen Seiteninformationen auf.
Grund hierfür ist der beschränkte Zugriff als freier Nutzer auf die WikipediaAPI. Gibt es als Seiteninformationen lediglich einen Titel, eine PageID und die Anzahl der Subartikel zu berücksichtigen, so stellen die anderen zu parsenden Daten potenziell sehr lange Listen dar. Das Abfragen dieser Listen kann als unregistrierter Benutzer der Wikipedia wie im unten anhand von Links gezeigten Beispiel nur in 10er Schritten erfolgen:

```
<query>
<normalized>
  <n from="Germany_at_the_2012_Summer_Olympics"
    to="Germany at the 2012 Summer Olympics"/>
</normalized>
<pages><page pageid="28580455" ns="0" title="Germany at the 2012 Summer Olympics">
 <links><pl ns="0" title="2012 Summer Olympics"/>
      <pl ns="0" title="2012 Summer Olympics medal table"/>
      <pl ns="0" title="ASSECO Resovia Rzeszow"/><pl ns="0" title="Abdelhafid Benchabla"/>
      <pl ns="0" title="Adam Okruashvili"/><pl ns="0" title="Adam Skrodzki"/>
      <pl ns="0" title="Adelheid Morath"/><pl ns="0" title="Adrian Crisan"/>
      <pl ns="0" title="Afghanistan at the 2012 Summer Olympics"/>
      <pl ns="0" title="Agnieszka Radwanska"/>
 </links>
 </page>
</pages>
```

```
  </query>
  <query-continue>
    <links plcontinue="28580455|0|Alaaeldin_Abouelkassem"/>
  </query-continue>
</api>
```

Aufgrunddessen wurde ein rekursiver Aufruf dieser Funktionen nötig, welcher den Continue Links (hier plcontinue) folgt und somit alle benötigten Daten erfassen kann. Nach erfolgreicher Erstellung des Korpus, sollte dieser exportiert werden:
Für den Korpusexport wurde auf einen FileWriter zurückgegegriffen, welcher die von Object überschriebene toString() Methode des Artikelobjekts für die Ausgabe verwendet. Der Export des Korpus enthält die folgenden Elemente:

ID	Title	Timestamps	Categories	Subarticles	Links
36665005	Rachel Bragg	4	[Category:Living people....]	0	[Volleyball....]

Timestamps weist hier die Gesamtanzahl alle Bearbeitungen des Artikels auf, die jemals vorgenommen wurden.

3.1.2 Ausgabe der Revisionen

Durch den Aufbau des Korpus wurden uns die Bearbeitungsdaten (Timestamps) der Artikel geliefert. Diese mussten zum Export in einen zeitlichen Zusammenhang gestellt werden. Zu diesem Zwecke wurde eine Map von Datum auf Bearbeitungen für jeweils alle Artikel im Korpus erstellt. Bei dem Einleseprozess der Revisionen wurden die Bearbeitungseinträge des aktuellen Datums erhöht. Die Map stellte somit die Bearbeitungen aller Artikel im Zusammenhang der Olympischen Spiele nach Datum sortiert dar. Für die korrekte Darstellung der Daten in Matlab wurden auch die Tage an denen keine Bearbeitungen im Zeitintervall stattfanden mit 0 in die Map aufgenommen.

```
while(currentdate.before(this.enddate)){
    currentdate.setTime(currentdate.getTime()+Olympia.SECONDSOFDAY);
    if(!this.getRevisions().containsKey(this.matlabdateformat.format(currentdate))){
      this.getRevisions().put(this.matlabdateformat.format(currentdate),0);
    }
}
```

Die Ausgabe der Revisionen erfolgte in 2 Txt-Dateien, von denen die erste, die Keys der Map (Tag 0-X) und die zweite die Values der Map, d.h. die Bearbeitungen an diesem Tag darstellt. Nach Import der beiden Vektoren in Matlab, kann ein Vergleich durch Plotten erfolgen.

3.1.3 Implementierung der Relationen

Um den RDF-Graphen zu exportieren, mussten nun die Entitäten der Dokumente identifiziert und Beziehungen zwischen ihnen hergestellt werden. Dafür wurde nach dem im vorherigen Kapitel beschriebenen Schema vorgegangen. Die benötigten Beziehungen wurden als Maps in der Klasse RDFData abgebildet.

```
public class RDFData implements RDFDataAPI{
      private final transient Map<String,Artikel> sport;
      /**Map of sports with their participating nations.*/
      private final transient Map<String,Set<String>> sportToNation;
      /**Map of sports with their participants.*/
      private final transient Map<String,Set<String>> sportToSportsmen;
      /**Map of sportsmen who have achieved gold medals including the amount.*/
      private final transient Map<String,Integer> sportsmenToGold;
      .....
}
```

Für den geforderten Export der Nationen, Sportler und Sportarten wurden bereits bei der Korpuserstellung nach o.g. Kriterien Maps der Entitäten auf die entsprechenden Artikel angefertigt. Anhand dieser Basisinformationen wurden die Beziehungen ermittelt: Die Methode createRelationships implementiert die Vorgehensweise:

```
for(Artikel sportsman:this.sportsmen.values()){
//Fuer alle Artikel
            boolean unknown =true;
            for(String link:sportsman.getLinks()){
//Pruefe Links
                  for(String nation:this.nations.keySet()){
                        if(link.equals(nation)
                  && sportsman.getBacklinks().contains(nation+RDFData.SUMMEROLYMPICS)){
                        //Fuege in die Relation Nation zu Sportler/Sportler zu Nation ein
                        }
                  }
            }
            if(unknown){  //Wenn wir noch keine Nation gefunden haben
                  for(String backlink:sportsman.getBacklinks()){
                        for(String nation:this.nations.keySet()){
                              if(backlink.equals(nation+" at the 2012 Summer Olympics")){
                                    //Fuege in die Relation Nation zu Sportler hinzu
                              }
                        }
                  }
            }
            if (unknown){
               //Ordne die Nation Unknown zu
            }
            for(String category:sportsman.getCategories()){
               for(String sport:this.sport.keySet()){
                  if(sport.substring(0,3).equals(category.substring(9,12))){
            //Fuege Sportler zur Sportart hinzu wenn die ersten
            //3 Buchstaben der Sportart/Category matchen
                  }
               }
            }
```

Die Relationen Sportler zu Sport und Sport zu Sportler können sich aus den nun ermittelten Relationen errechnen.

3.1.4 Medaillen

Die Medaillen wurden von den Übersichtsseiten der Sportarten extrahiert und nach einigen Anpassungen mit einem SAXParser geparst. Der DefaultHandler zum Parsen wurde in der Klasse MedalHandler implementiert. Die wichtigsten Funktionen der Klasse bestehen aus:

1. Auffinden der Tabellenstrukturen der Medaillen über das Keyword ”Event”
2. Abgrenzung anderer ”Event” Tabellen ohne Medaillen
3. Indentifikation der Sportler und deren Medaillen
4. Zuordnung der Sportler zu den Medaillen in der Datenstruktur

Die Tabellen der Medaillen sind in Wikipedia übersichtlich auf den Seiten der Sportarten aufgelistet. Allerdings haben die Tabellen bis auf die Titelspalten ”Event”,”Gold”,”Silver”, ”Bronze” kein einheitliches Format. So kann es beispielsweise vorkommen, dass sich die Spalte Gold in zwei Unterspalten mit dem Sportler und der Rekordzeit aufspaltet. Um dennoch die Sportlernamen extrahieren zu können, wurde die Spaltenanzahl der ersten Tabellenzeile bestimmt und einheitlich aufgeteilt.
Anschließend wird jeder geparste String in den identifizierten Tabellen mit der Sportlerliste abgeglichen. Wird ein Sportler identifiziert, so wird er in die entsprechende Medaillenrelation eingefügt.

```
public void addData(String chars){
        if(!chars.contains(”(”) && !chars.contains(”\n”) && !””.equals(chars) && !” ”.equals(chars) && !chars.matches(”[0-
            System.out.println(”Sportsman or Nation: ”+chars);
            if(this.rdf.getSportsmen().containsKey(chars)){
                this.rdf.addSportsmanToMedal(chars, this.medalcounter);
            }
            }
}
```

3.1.5 RDF-Format

Aus den ermittelten Daten wird das im vorherigen Kapitel definierte RDF-Format mit der Funktion rdfExport Das RDF-Format ist nach dem folgenden Schema aufgebaut:

```
<?xml version=”1.0” encoding=”utf−8”?>
<!−− Datei XSDTest.xsd −−>
<xsd:schema
xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
<xsd:complexType name=”nation”>                    <!−− Nation Type Definition −−>
<xsd:element name=”tt:isrepresented” type=”nationsportsmenlist”> <!−− Sportsmen of Nation −−>
<xsd:complexType name=”nationsportsmenlist”>
<xsd:sequence minOccurs=”0” maxOccurs=”unbounded”>
<xsd:element name=”rdf:Description” type=”xsd:string”>
<xsd:attribute name=”rdf:about” type=”xsd:string”/>
</xsd:element>
```

```
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="tt:participates" type="xsd:string">  <!-- Sports of Nation -->
<xsd:complexType>
<xsd:sequence minOccurs="0" maxOccurs="unbounded">
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:complexType>
<xsd:complexType name="sportsman">                    <!-- Sportsman Type Definition -->
<xsd:element name="tt:competing" type="xsd:string">
<xsd:sequence minOccurs="0" maxOccurs="unbounded">      <!-- Sports of Sportsmen -->
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:element>
<xsd:element name="tt:nation" type="xsd:string"/>
<xsd:element name="tt:gold" type="xsd:nonNegativeInteger"/>
<xsd:element name="tt:silver" type="xsd:nonNegativeInteger"/>
<xsd:element name="tt:bronze" type="xsd:nonNegativeInteger"/>
</xsd:complexType>
<xsd:complexType name="sport">                       <!-- Sport Type Definition -->
<xsd:element name="tt:nation" type="xsd:string">
<xsd:sequence minOccurs="0" maxOccurs="unbounded">      <!-- Nation of Sport -->
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:element>
<xsd:element name="tt:sportsman" type="xsd:string">
<xsd:sequence minOccurs="0" maxOccurs="unbounded">      <!-- Sportmen of Sport -->
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:element>
</xsd:complexType>
<xsd:complexType name="bronze">                      <!-- Bronze Medals Definition -->
<xsd:element name="tt:sportsman" type="xsd:string">
<xsd:sequence minOccurs="0" maxOccurs="unbounded">
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:element>
</xsd:complexType>
<xsd:complexType name="gold">                        <!-- Gold Medals Definition -->
<xsd:element name="tt:sportsman" type="xsd:string">
<xsd:sequence minOccurs="0" maxOccurs="unbounded">
```

```
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:element>
</xsd:complexType>
<xsd:complexType name="silver">                    <!-- Silver Medals Definition -->
<xsd:element name="tt:sportsman" type="xsd:string">
<xsd:sequence minOccurs="0" maxOccurs="unbounded">
<xsd:element name="rdf:Description" type="xsd:string">
<xsd:attribute name="rdf:about" type="xsd:string"/>
</xsd:element>
</xsd:sequence>
</xsd:element>
</xsd:complexType>
<xsd:element name="rdf:RDF" type="xsd:string">
<xsd:element name="rdf:Description" type="nation"/>     <!-- M Nationen -->
<xsd:element name="rdf:Description" type="sportsman"/>  <!-- N Sportler -->
<xsd:element name="rdf:Description" type="sport"/>      <!-- O Sportarten -->
<xsd:element name="rdf:Description" type="bronze"/>     <!-- B Bronzemedaillengewinner -->
<xsd:element name="rdf:Description" type="gold"/>       <!-- G Goldmedaillengewinner -->
<xsd:element name="rdf:Description" type="silver"/>     <!-- S Silbermedaillengewinner -->
</xsd:element>
</xsd:schema>
```

Dieses Schema soll nur als Veranschaulichung verstanden werden. Es ist produktiv nicht einsatzfähig oder getestet worden. Dennoch zeigt es die Eigenschaften des Exportformates recht gut. Der Export bildet die in RDFData gegebenen Relationen wie weiter oben beschrieben ab. Die Beziehungen bleiben dabei gewahrt und werden in RDF überführt.

3.2 Dokumentation

Für die Dokumentation der Quelltexte wurde JavaDoc verwendet. Das generierte Javadoc finden Sie in den Projektdateien[1]. Es wurden sämtliche Methoden, Klassen, Interfaces und Attribute, sowie Konstanten der Klassen dokumentiert und sofern als notwendig erachtet weiter Kommentare an ausgewählten Stellen der Methoden ergänzt.

3.3 Anwenderdokumentation

Die Anwendung ist in Java geschrieben und erfordert zur Ausführung eine Java Virtual Machine 1.6 oder höher. Zur Installation muss lediglich das JAR-File kopiert und mit dem Befehl

```
java -jar Olympia.jar 16 01-02-2004 01-07-2012 //Threads + Start/Enddatum
java -jar Olympia.jar 01-02-2004 01-07-2012    //Start/Enddatum
java -jar Olympia.jar 16                       //Threads
java -jar Olympia.jar
```

[1] doc/index.html

ausgeführt werden.
Hierbei kann über Parameter das Start- und Enddatum für die Auswertung sowie die Anzahl der Threads zur parallelen Bearbeitung der Seiten angewählt werden. Werden keine Parameter angegeben, arbeitet das Programm mit 2 Threads und ermittelt das Datum des ersten Artikels von Olympia 2012 und das Datum des letzten Artikels als Parameter. Weiterhin wird für die Ausführung des Programmes, speziell mit Erhöhung der Threads eine gute Internetverbindung empfohlen.

3.3.1 Ergebnisse

Nach Ende der Ausführung des Programmes finden sich die Ergebnisdateien im Unterverzeichnis "out" des Programmes. Sie beinhalten:

- korpus.txt - Die exportierte Version des Korpus
- matlab1.txt - Die Range der Revisionstage
- matlab2.txt - Die Editierungen an den jeweiligen Revisionstagen
- nation.txt - Die Liste der Nationen
- rdfExport.rdf - Der Export der RDF-Relationen
- sport.txt - Die Liste der Sportarten
- sportsmen.txt - Die Liste der Sportler

Hierbei kann die RDF-Datei beispielsweise in einem RDF-Visualisierer wie Gravity[2] anzeigen lassen. Die Dateien matlab1.txt und matlab2.txt können als Matrizen in Matlab importiert werden und graphisch geplottet werden. Sämtliche andere Dateien dienen der Veranschaulichung und sind auf kein konkretes Programm hin optimiert worden.

4 Analyse des erstellten Korpus

Die Analyse des erstellten Korpus erfolgt unter verschiedenen Gesichtspunkten. So wurde zunächst Wert auf eine Gesamtübersicht der ermittelten Ergebnisse gelegt und im Weiteren auffällige Bestandteile der Übersicht näher betrachtet.

4.1 Gesamteindruck

Die vorige Abbildung zeigt die Gesamtübersicht der Revisionen beginnend mit dem Startdatum der Erstellung der Seite Olympische Spiele 2012. In der Grafik bezeichnen wir diesen Tag als Tag 0. Die darauffolgenden Tage bis zur letzten aufgezeichneten Änderung des Korpus sind aufsteigend nummeriert. Auffällig bei der Analyse des Ergebnisses ist der sprunghafte Anstieg der Revisionen in dem Abschnitt des Jahres 2012, in dem die Olympischen Spiele stattfanden. Eine Steigerung auf nahezu 800 Revisionen ist hier zu verzeichnen.

[2] http://semweb.salzburgresearch.at/apps/rdf-gravity/download.html

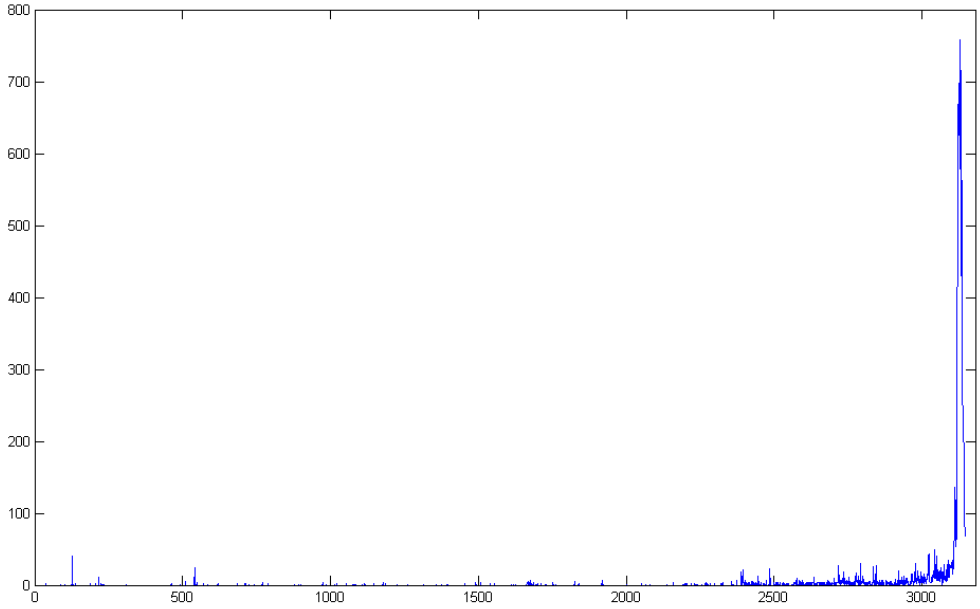

Abbildung 3: Lineare Darstellung aller Revisionen

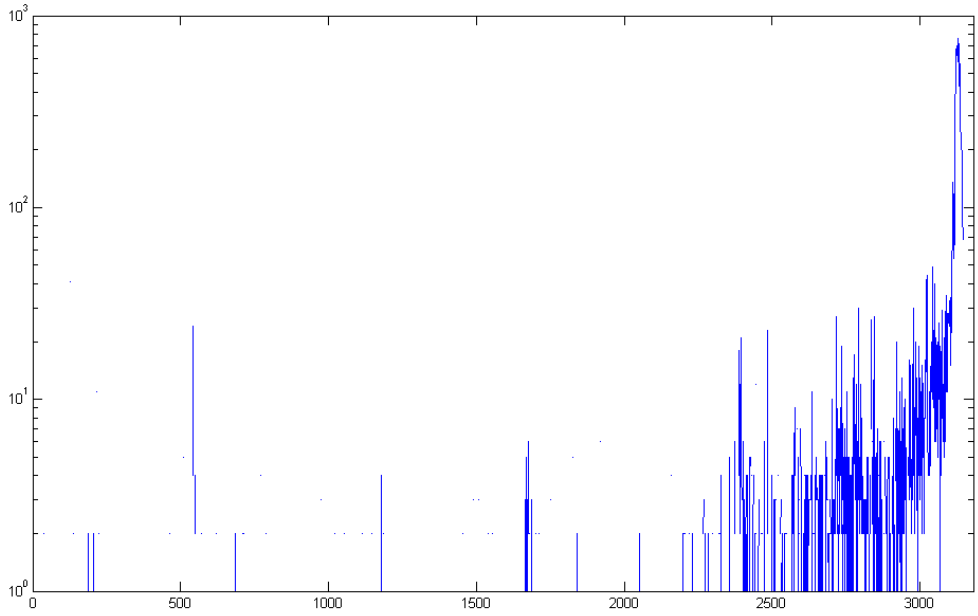

Abbildung 4: Logarithmische Darstellung aller Revisionen

Diese Grafik zeigt die Auswertung zur Verdeutlichung der Unterschiede in einem logarithmischen Graphen. Es lässt sich gleiches beobachten, allerdings werden einige auffällige Häufungen von vielen Revisionen sichtbarer. Diese Revisionen lassen sich Ereignissen in dieser Zeit zuordnen, von denen im Weiteren noch die Rede sein wird.

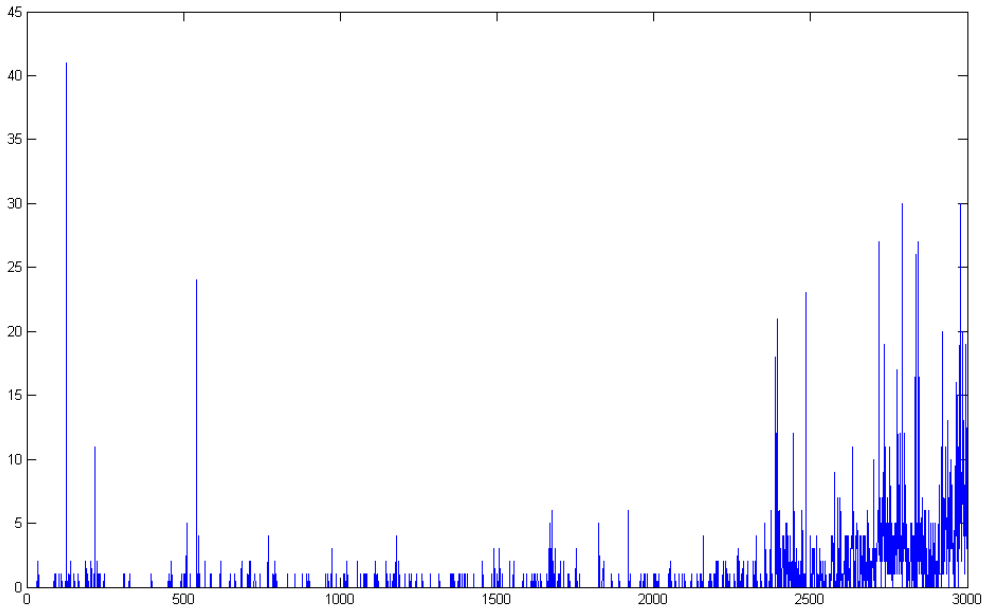

Abbildung 5: Revisionen ohne das Jahr 2012

Durch das Stattfinden der Olympiade in einem Zeitraum in 2012, fanden sich dort exorbitante Häufungen von Revisionen. In diesem Bild wurde das Jahr 2012 entfernt um einen Überblick auf die Jahre davor zu bekommen.

4.2 Events

Aus der Revisionsübersicht ließen sich einige besonders auffällige Peaks, an denen besonders viele Revisionen erstellt wurden herausgreifen und einordnen. Diese sollen hier in ihrem zeitlichen Zusammenhang dargestellt werden:

4.2.1 Auslosungsverfahren

Das Auslosungsverfahren für den Austragungsort der Olympischen Spiele 2012 startete im Jahre 2004. Dies stellt den ersten Peak des Revisionsgraphen dar:

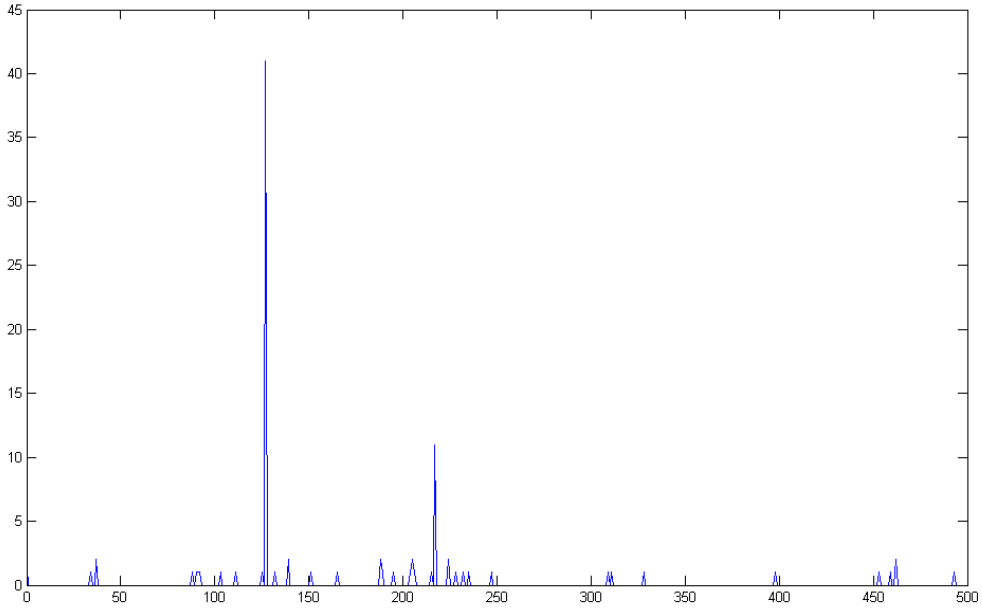

Abbildung 6: Vorstellung der Kandidaten

Deutlich lässt sich hier ein Anwachsen der Revisionen auf 40 erkennen.

4.2.2 Bekanntgabe des Austragungsortes

Bei Bekanntgabe des Ausstragungsortes am 6.Juli 2005 ließ sich ein deutlich sichtbarer Anstieg der Revisionen auf den Wert 24 erkennen:

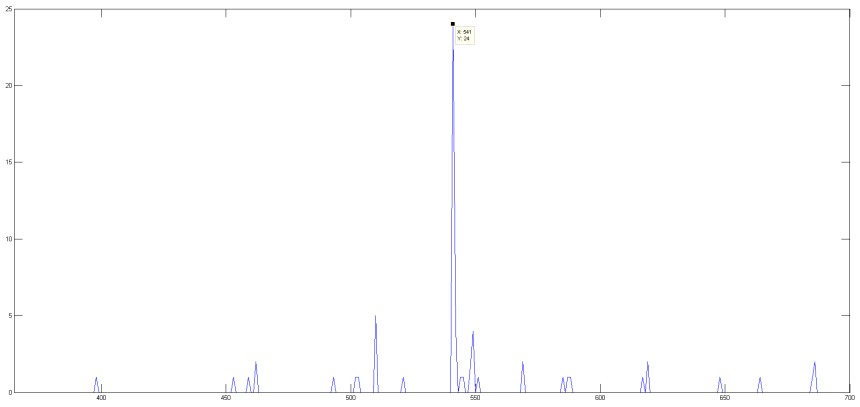

Abbildung 7: Revisionen bei Verkündung des Austragungsortes

4.2.3 Olympia in Peking 2008

Interessante Parallelen fanden sich zudem während der Olympischen Spiele in Peking 2008. Der im folgenden dargestellte Revisionsgraph zeigt auch hier eine erhöhte Editierfrequenz an, jedoch in einem weitaus geringeren Maße .

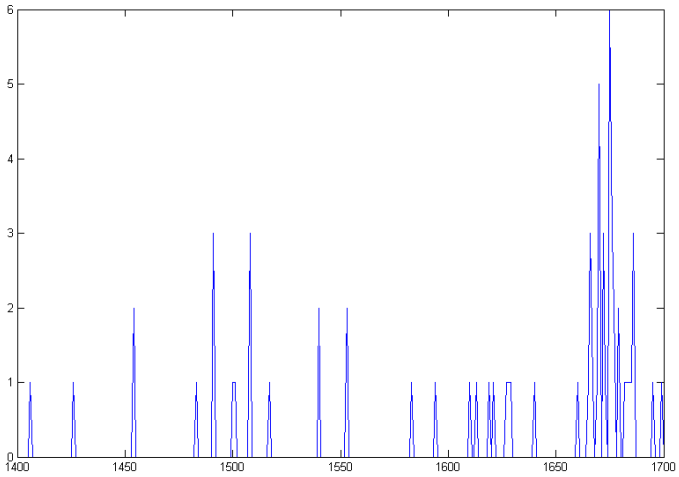

Abbildung 8: Revisionen während Olympia 2008

4.2.4 Die Jahre 2010/2011

Im nächsten Abschnitt ist der Verlauf der Revisionen in den Jahren 2010 und 2011 dargestellt. Auch hier lässt sich eine erhöhte Editieraktivität in Anbetracht der Vorbereitungen zu den Olympischen Spielen feststellen.

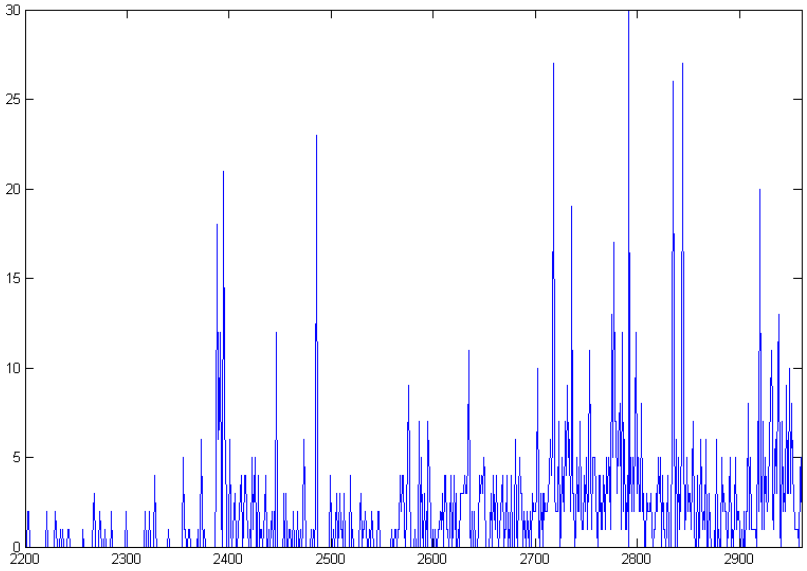

Abbildung 9: Revisionen in den Jahren 2010 und 2011

5 Kontrolle des RDF-Exports

Die resultierenden Triples bieten leider keine exakte Repräsentation aller Zusammenhänge, wie sie während der Olympiade 2012 vorherrschten, aber dennoch eine gute Repräsentation der Zusammenhänge in der Wikipedia zu dem Thema, was letztendlich auch der Fokus der Aufgabe war. Wie bereits im Lösungsweg angesprochen sorgte die Sportart Beachvolleyball für Probleme da es offiziell kein Olympisches Großevent ist, sondern nur eine Form des Volleyballs. Keiner der Beachvolleyballer in der Wikipedia wird jedoch mit "Volleyball" in Verbindung gebracht, sodass eine Auswertung in diesem Fall nicht möglich war.

Artikel Stubs sorgen außerdem dafür, das nicht garantiert werden kann, das die Sportler-zu-Nation Relation immer korrekt ermittelt wird. Basierend auf Stichproben scheinen sie allerdings überwiegend korrekt zu sein.

Beispielhaft soll dies an der Nation American Samoa deutlich gemacht werden:
Die folgenden in RDF ermittelten Daten

```
<tt:isrepresented rdf:parseType=”Collection”>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Anthony_Liu_(judoka)”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Ching_Maou_Wei”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/’Elama_Faatonu”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Megan_Fonteno”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Nathaniel_Tuamoheloa”>
</rdf:Description>
</tt:isrepresented>
<tt:participates rdf:parseType=”Collection”>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Athletics_at_the_2012_Summer_Olympics”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Judo_at_the_2012_Summer_Olympics”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Swimming_at_the_2012_Summer_Olympics”>
</rdf:Description>
<rdf:Description rdf:about=”http://en.wikipedia.org/wiki/Wrestling_at_the_2012_Summer_Olympics”>
</rdf:Description>
</tt:participates>
</rdf:Description>
```

stimmen mit den Daten die auf der Seite
http://en.wikipedia.org/wiki/American_Samoa_at_the_2012_Summer_Olympics
zu sehen sind überein

Ein Fehler der bei wiederholten Stichproben zu finden ist beruht auf der Inkonsistenz der Wikipedia Daten. In allen Kategorien, Sportlerseiten und Links wird Großbritannien als ” Great Britain ” bezeichnet. Vereinzelte Sportler stammen jedoch aus dem ”United Kingdom”, das als solches aber nicht an der Olympiade teilgenommen hat, aber auch kein einfaches Synonym ist, das durch Überprüfung der Redirect-Informationen abgefangen werden könnte. Letztendlich führt es dazu das er Afghanistan zugeordnet wird da dies die erste Seite ist von der er zumindest einen Backlink besitzt. Dies ist eindeutig ein Fehler, aber ohne sein Land als Sonderfall zu berücksichtigen wohl nicht abzufangen.

6 Zusammenfassung

Zusammenfassend lässt sich sagen das sich zeigen ließ, wie ein sportliches Großereignis wie die Olympischen Spiele , im Vorfeld, aber insbesondere während der Spiele, einen deutlichen Fußabdruck in der Wikipedia hinterlässt. Die Aktivitätsspitzen scheinen dabei nicht willkürlich zu sein, sondern liegen meist zeitnah an einem oder mehreren direkt damit in Verbindung stehenden Ereignis(sen).

Die Daten in der deutschsprachigen Wikipedia beschränkten sich leider größtenteils auf deutsche Athleten, weswegen wir die englische Wikipedia nutzten um unseren Corpus zu erstellen. Zahlreiche Stubs und das Fehlen einer einheitlichen Struktur, trotz der gegebenen Templates, führte leider zu einem komplexeren (und damit weniger allgemeinen) Entscheidungsbaum, welcher zur Kategorisierung der Sportler/Nationen/Sportarten/Medaillen sowie ihrer Relationen, notwendig war. Der resultierende RDF ist dafür, gemessen an den Informationen in der Wikipedia, sehr akkurat.