Abschluss Arbeit

Neo4j und..

Asmaa Haja	Ilhem Bouzir	Josseph
348758	350778	330187

6. Juli 2015

Muss geaendert werden: TU Berlin – Fakultät IV
Institut für Wirtschaftsinformatik und Quantitative Methoden
Fachgebiet Informatik und Gesellschaft
Prof. Dr.-Ing. Frank Pallas
Max Ulbricht

Inhaltsverzeichnis

Abbildungsverzeichnis

Listings

1 Abstract

Dieser Bericht beschreibt die Erstellung eines knowledgegraphs mit Hilfe von Neo4j. Dieser Graph repräsentiert verschiedene Deutsche städte ,deren Sehenswürdigkeiten und deren Entfernungen voneinander. Das Projekt fand im Rahmen der Veranstaltung DBPRO an der TU Berlin statt.

2 Einleitung

Im Bereich Big Data fallen zum Beispiel bei Texten sehr große Daten-mengen an. Sinnvollerweise sollten solche Text Informationen auf einer übersichtlicherer und anschaulischerer Weise dargestellt werden können, so dass anhand einer bestimmten Repräsentation der Daten aus diesem Text gewonnenwerden. Diese erlaubt eine leichtere Suche nach bestimmten Informationen, ohne dass dabei wichtige Daten verloren gehen. Ziel des Projektes war es eine Knowledge Graph mit Hilfe der Graphendatenbank Neo4j zu erstellen basierend auf bestimmte rausgesuchte Daten.Mit der Graphenbasierte Abfragesprache sollen dann benötigte Informationen aus einer Wikipedia Informationen basierte Linked Open Data Version (DBpedia) extrahiert werden können und in einem bestimmten Format zu Neo4j übergen werden, so dass diese die Daten im strukturierten Graphen Form automatisch bringt. Das Projekt fand im Sommersemester 2015 an der TU Berlin im Rahmen der Veranstaltung DBPRO statt. Das Projektteam bestand aus Asmaa Haja, Ilhem Bouzir und Josseph, welche durch Dr. Holmer Hemsen betreut wurden.Das Projekt umfasste drei Meileinsteine. Zunächst fand im 1. Meilenstein eine Einarbeitung in das Thema Knowledge Graph, Neo4j und SPARQL statt. Als nächstes wurde dann konkrete Beispiele für Tourism in Deutschland mit SPARQL erstellt. Im 3. Meilenstein wurde aus der zuvor erstellten Abfragen mit Hilfe von Neo4j ein Knowledge Graph erstellt.Diese Arbeiten werden in diesem Bericht nach einer genauen Identifizierung der Aufgaben näher erläutert. Abschließend folgen ein Kapitel zur Evaluation und eine kurze Zusammenfassung der Projektergebnisse.

3 Aufgaben

Als nächstes gliedern wir die Erarbeitung des Projektes in drei Mailsteine, welche die zu erreichenden Ziele im laufe des Projektes beinhalten:

- Meilstein 1
  - Eingewöhnung Knowledge Graphen
  - Eingewöhnung Neo4j und Cyper-Sprache
  - Eingewöhnung Sparql
- Meilstein 2
  - Eingewöhnung Dbpedia

  - Eingewöhnung OWL, RFD, RFDS
  - Anfragen an die strukturierten Internetwebseiten über die Sparql-Interface
- Meilstein 3
  - Automatizierung der Sparqlanfragen über Java.
  - Automazierte Erstellung des Datenbakes über Java.
  - Erstellung einer Schnittstelle(GUI) für die Benutzer um Anfragen an die Datenbank zu ermöglichen

4 Umsetzung

Die bearbeitung der einzelnen Ziele werden demnächst näher erläutert.

4.1 Knolwledge Graphen

knowledge Graph wurde im 1982 von Hoede and Stokman entwickelt. Sie hatten den Ziel Wissen aus medizinischen und soziologischen Texte zu extrahieren um Expertensysteme zu erhalten und dadurch neue Methoden um das Wissen zu repräsentieren. Knowledge Graphen gehört zu der Kategorie semantischen Netze. [6]

Wissen kommt im Netzwek im strukturierter als auch im unstrukturierter Form vor. Außerdem kann das in deutlicher als auch undeutlicher Form gestaltet sein. Das fürht dazu, dass der nach Information suchenden Netzwerknutzer zu sowohl sinnvollem als auch unbrauchbarem Wissen kommt. Um der Neztwerknutzer einen sinnvollen Überblick über das Wissen im Nezt zu erschaffen, Wird das in sctrukturierter Form aufgebaut. Auf diese Weise kommen wir zur Knowledge Graphen. Was nicht anderes ist als eine Art von semantischen Netzwerke, wo aus der Analysephase von texten zu einer Liste von Konzepte führt. Aus diese Listen werden bennaten Knoten erzeugt und letztendlich verbindet man sie anhand von Relationen, welche die Beziehungen zwieschen Konzepte(Knoten) darstellt und zu der Strukturierung des Wissen hinführt.[4]

4.2 Neo4j und Cypher

Neo4j ist ein open-source NoSQL graph Database, welche im Java und Scala implementiert wurde und von Neo Technology unterstützt wird. Die entwicklug began im 2003 und der freie Zugang ist seit 2007 veröffentlicht worden. Diese Technologien findet Anwendungen im folgenden Bereiche: Matchmaking, Netzwerk-Management, Software-Analytik, wissenschaftliche Forschung , soziale Netzwerke und mehr. Um die Arbeitsweise von Neo4j zu verstehen brauchen wir die Begriffe "Property Graph Databaseünd "NoSQL"kennen lernnen. Ein Property Graph ist, wie der Name einwenig verratet, ein aus Knoten und kanten Bestehendes Graph.welsches sich von einem normalen Graph dadurch unterscheidet, dass sowohl Knoten als auch Kanten Eigenschaften besitzen können. Und der Begriff "NoSQlïst die Abkürzung für No Only SQL,diese Spracheanfrage umgeht

die Leistrungproblemen bei der Anfrage an relationalen Datenbanken in dem durch vernachlässigen von Consistency eine Verbeserung von Availability und Partitioning statt findet. Das fürht dazu, dass die NoSQL Database sich besser skalieren lassen.(Die Abbildung 1 zeigt wie Daten mit Neo4j dargestellt werden

Cypher ist eine deklarative Sprache, welche von Neo4j benutzt wird, Um Anfragen an die Datenbank zu ermöglichen. Ein wichtiger Vorteil ist, dass beim Stellen einer Frage an die Datenbank nur angegeben wird, wonach gesucht wird und nicht wie man es finden soll.[3]

Die Abildung 1 zeigt ein klienes Beispiel,wie Daten im Neo4j dargestellt werden und wie man mit der untenligenden Cypher-Anfrage auf die Knoten und die dazwischen liegende Relation zugreifen kann. Beide Knoten sind von Type City Und die Relation ist von Type Distance

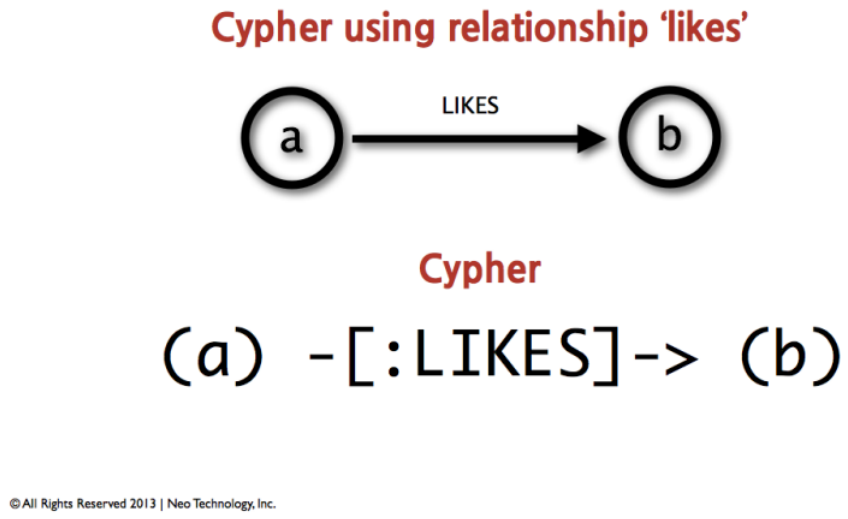

Abbildung 1: Query Beispeil [3]

4.3 Sparql

Sparql ist eine für relationale Datenbanken Anfragesprache, Welche im erste linie entwickelt worden ist um RDF Graphen anzufragen. Der Baustein für SPARQL Abfragen ist Basic Graph Patterns(BGP). SPARQL(BGP) ist eine Menge aus triple patterns, welche im Abschnitt RDF näher erläutert wird. Um komplexe Anfraguen aufzubauen, benutzt man die Operatoren Select, Optional, Union, und Filter. [5]
Da RDF-Dokumente eine Menge von RDF-trippeln ist. Verwendet Spaqrl eine ähnliche strukturierung. d.h zum Aufbau eine Anfraguemuster verwendet man Tripeln der Form (Subjekt,Predikat,Objekt), Wobei das Beenden des Tripeln duch einen Pukt gekennzeichnet wird. Außerdem kann man für die Teile eines Tripels variablen verwenden,

welsche mit "?"gekennzeichnet werden. In Abbildung 3 ist eine Beispielanfrague zu erkennen.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?city ?cprop ?country ?label
WHERE { ?city rdf:type dbpedia-owl:City ;
              rdfs:label ?label ;
              dbpedia-owl:country ?country ;
              dbpprop:countryCode ?cprop
}
```

Der SELECT Klausel enthält die Variablen, welche nach ausführung der Anfrague belegt worden sind. Im der WHERE Klausel wird der Aufbau des gewüchten Graphs beschrieben.Die komplette Sprachsyntax, sowie alle Sprachmittel, sind in [W3C SPARQL] zu finden.

4.4 Dbpedia

Im rahmen dieses Projekts benutzen wir Dbpedia als prinzipal Quelle um informationen aus Wikipedia zu extrahieren.

DBpedia ist eine Gemeinschaftsbemühung um strukturierter Information aus Wikipedia zu extrahieren. Es handelt sich um eine sammliung von Quellen, welche Wissen in verschiedenen Sprachen zur Verfügung stellt.Das zum extrahierenden Wissen kommt im struktuierter Form vor, welche durch Ontologien definiert worden sind. Das DBpedia Project wurde im 2006 gestartet und hat mittlerweile große Interesse im Forschungsprojekte und in der Praxis geweckt[1]

4.5 OWL, RDF, RDFS

OWL, RDF und RDFS sind OntologieSprachen, die dazu dienen eine Ontologie zu beschreiben. Zuallererst erläutern wir kurz was unter dem Begriff Ontologie zu verstehen ist. Dieser bedeutet eine Wissenvermittlung, die eine Bestimmte Domäne beschreibt. [2] Ontology ist ein Bestandteil von DBpedia. Diese kommt auch Komplexe Anfragen entgegen indem sie trotz der Komlexität genaue und Präzise Antworten liefert. DBpedia Ontology umfasst Klassen, Eigenschaften und Instanzen. Der Aufbau einer Ontology wird es die untere Abbildung zeigen. Diese hat 2 Klassen City und Attraction und die beiden dazugehöigen Instanzen Berlin und Fernsehturm. Die beiden klassen verbindet außerdem die has Relation.

Damit eine Ontology von Maschinen erkennbar wird ist ein Formaler Aufbau benötigt. Deswegen wird Resource Description Framework oder kurz RDF verwendet. Dies ist eine

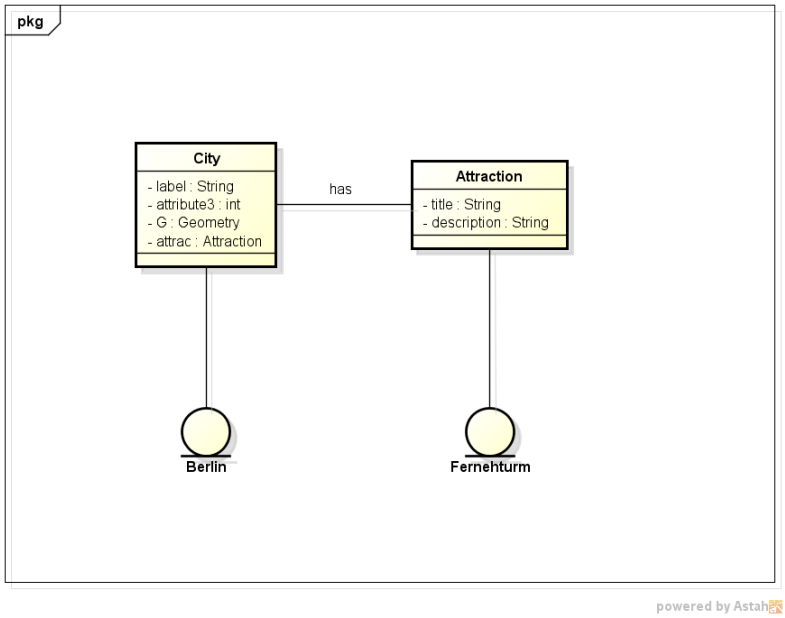

Abbildung 2: Ontologie

Triple basierende Sprache und besteht aus Subject Prädikat und Object als Grundlage diese Elemnte werden wie es in der Abbildung 3 zu sehen ist verwendet.

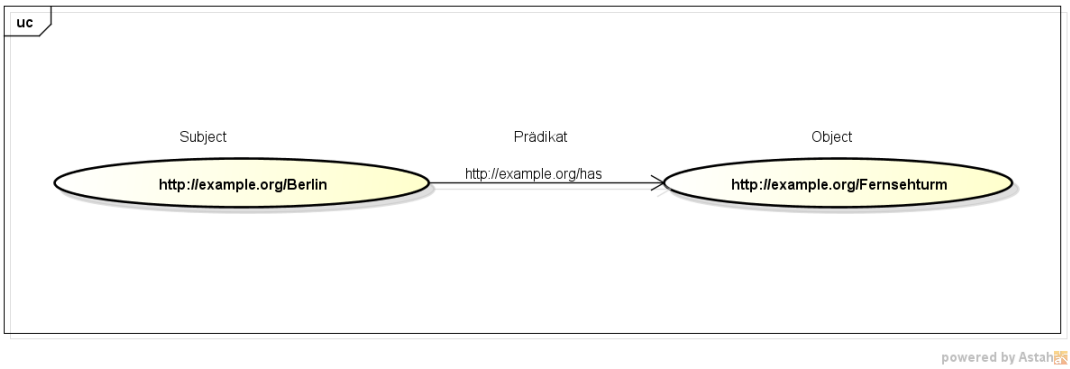

Abbildung 3: RDF

RDF Schema oder kurz RDFS präsentiert eine Erweiterung bei dem Vokabular vom RDF das die verschiedenen Klassenhierarchien ausdrücken lässt. Mit der RDF Verwendung sind die Ressourcen minimal typisiert deswegen fügt RDFS zu den RDF Konstrukten sowie Klassen und Properties weitere Typen hinzu. Außerdem kann anhand des RDFS die Spezialisierung und Generalissierung spezifiziert zwischen den Klassen werden unter der Verwendung von Subklassen was garnicht mit dem RDF Vokabularien darstellbar ist. Letztendlich bietet sich noch eine weitere Sprache Web Ontology Langua-

ge kurz OWL die aus noch Komplexerer Vokabularien entseht und damit komplexerer Sachverhälte modellieren erlaubt. Um die eine Ontology zu formalisieren basiert sich die OWL auf RDF und XML Syntax und verwendet auch RDFS Elemente. Zumal hat OWL andere Eigenschaften sowie Mengen Relation Operationen und Kardinalitätenbeschrängungen. Diese extra Funktionalitäten stärken die Ausdrucksmöglichkeiten und die Beschreibungsfähigkeiten von den zu formolierenden Aussagen.

4.6 Anfragen an die strukturierten Internetwebseiten über die Sparql-Interface

Um Daten aus Internetseiten wie DBpedia zu extrahieren benutzen wir die Snorql Interface, welche unter der LInk http://dbpedia.org/snorql zu erreichen ist. Abbildung 2 zeigt eine Sparql-Anfrague, welche die einige staädte von Deuschland angibt.

4.7 Automatizierung der Sparqlanfragen über Java

Damit wir unsere benötigte Tourism Informationen extrahieren und danach die automatisch an die Neo4j übergeben können, haben wir einen Java Code mit verschiedenen Funktionen implementiert. Diese sind bei der unteren Abbildung zu erkennen. Sodass am ende ein Knowledgegraph automatisch erstellt werden kann. Zuraller erst werden die Sparql Anfragen auf DBpedia ausgeführt sodass wir alle unsere Daten zugeliefert bekommen. Anhand der Attraction_Query() Methode werden dann die Attraktionen jeder Stadt in einer Liste gespeichert. Bei der Klasse Germany_Tourism werden dann diese erhaltene Daten geparst und in CSV Dateien gespeichert anhand der beiden saveAttraction() und saveDistance() Methoden wobei die zweite wird erst aufgerufen nachdem die calculateDistance() Funktion bereits ausgeführt wurde. Zuallerletzt werden denn diese Datendateien an die Neo4j übermittelt. Dieser liest denn diese Informationen aus und stellt sie in Form eines Graphes. Soweit hat man der Tourism Knowledge Graph, der die verschiedenen städten, deren zahlreichen Sehenswürdigkeiten und deren Entfernungen voneinander. Auf den Graph können dann belibige Cypher basierende Anfragen angewendet werden um gesuchte Informationen zu erhalten. Diese kann man dann mit Hilfe der Entwickelten GUI abfragen.

4.8 Benutzerschnittstelle(GUI)

Damit einem Nutzer sich informationen über die Tourismusmöglichkeiten im Deutschland angeben lässt, haben wir einen Schnittstelle(GUI) erzeugt. Welche durch einfacheres klicken sich bedinen lässt. Die Bedienungsschritte werden wir in den nächten Abchnitte beschreiben.

- Nach starten des Programs bekommt einen Benutzer das im Abbildung 3 gezeichnete GUI zu Gesicht. Durch klicken des Buttons Start wird der Datenbank generiert.

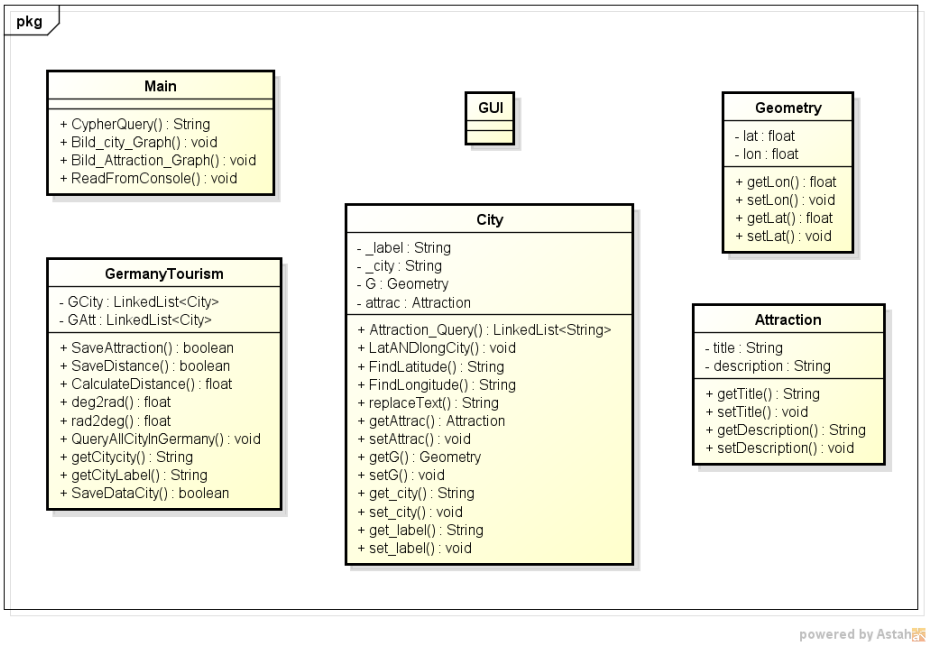

Abbildung 4: KlassenDiagram [3]

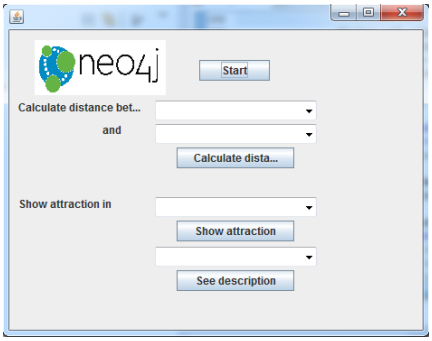

Abbildung 5: Hauptansicht

- Nach der Erstellung des DatenBanks werden Alle deutschen Städte im den Textfelder, die rechts von Calculate distance-,and- und Show atraction in-Labels aufgelistet, sodass der Nuzter per einfacheres klicken zwei städte auswählen kann und sich die Entfernug angeben lässt oder die Sehenwürdigkeiten eines Stadts aufgelistet angeben lässt.

- Falls für die Berechnung der Entfernung zwei Städte der selbe Stadt sowohl als start als auch als end Ziel gewählt wurde, erschient einen nachrichten Fenster, Wodurch der Kunde angedeutet wird, dass er zwei verschiedene Städte auswählen soll. Im Abbildung 7 wird dies gezeigt.

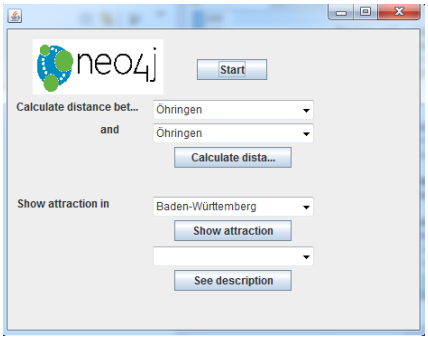

Abbildung 6: Datenbank wurde erzeugt

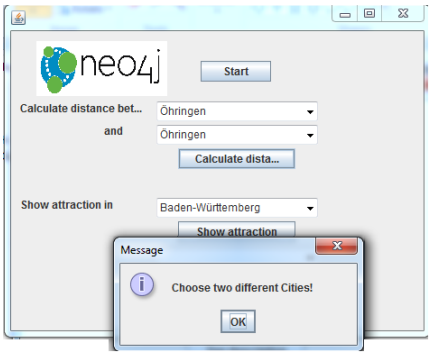

Abbildung 7: Auswahl von start und end City gleich gewählt

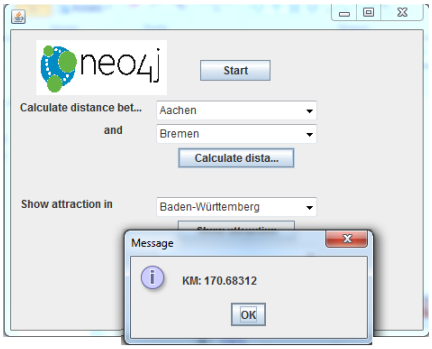

Abbildung 8: Ergebnis Bei der berechnung zwei verschiedene Städte

•

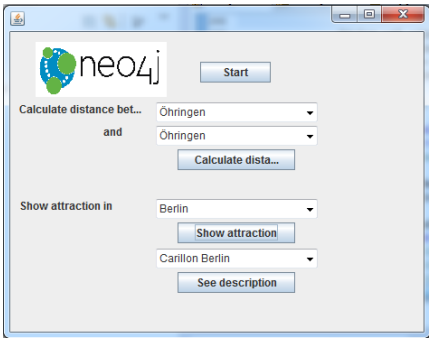

Abbildung 9: Hauptansicht

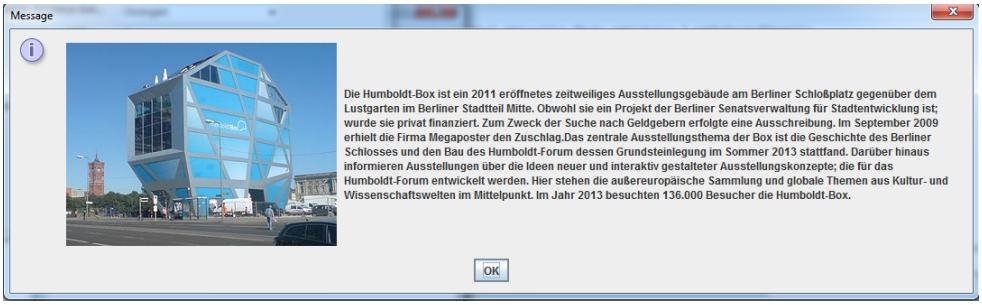

Abbildung 10: Hauptansicht

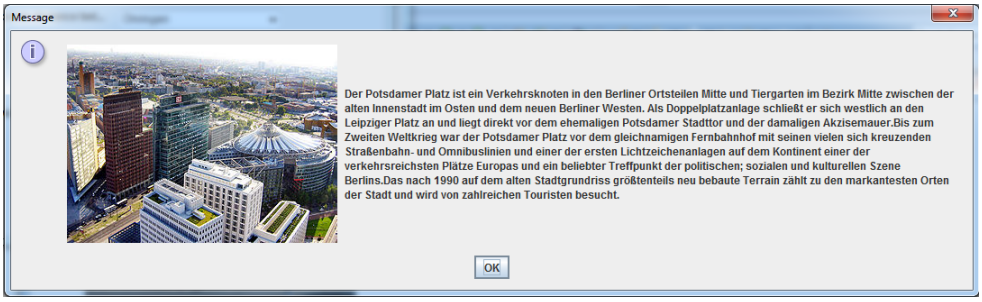

Abbildung 11: Hauptansicht

5 Anleitung

Dieser Abschnitt gibt einen kurzen Überblick über die Vorgehensweise beim Nutzen der entwickelten Knowledgegraphs und des implementierten Codes. Genauere Erläuterungen zu deneinzelnen Abschnitten befinden sich im Kapitel 4.

6 Evaluation

Die im Abschnitt 3 gesetzten Ziele wurden erfüllt. Mit Hilfe der DBpedia und den Sparql Anfragen Deutsche Städte Namen und verschiedene dazugehörige Tourism Informationen. Mitteles eines Java Codes und Neo4j wurden diese erhaltene Informationen, die Tourism in Deutschland inkorporieren im Form von Graphen dargestellt. Anschließend kann der Nutzer anhand eines Interfaces oder auch ohne nützliche gebrauchte Informationen Abragen. Er kann sowohl kürzester Strecke Information als auch Attraktionen einer Stadt abfragen. Die Nutzung DBpedia erleichtet die Arbeit enorm, allerdings sind die Informationen nicht unbedingt Vollständig gespeichert wodurch Probleme entstanden sind vorallem beim Städten auflisten und beim Verknüpfen von dieser mit den entsprechenden Sehenswürdigkeiten. Das entwickelte Java-Programm wurde unter Windows mit Java 7getestet.

7 Zusammenfassung

In der vorliegenden Arbeit wurde mit Hilfe der Neo4j Graphendatenbank durch Tourism in Deutschland zusammenhängende Informationen ein Knowledgegraph generiert. Anfragen an dieser Graph wurden mit Hilfe der Cypher Anfragesprache realisiert. Die Realisation erfolgte durch DBpedia basiertes Sparql Anfragen und einem Java-Modul, welches im Laufe des Prjekts implementiert wurde.

Literatur

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[2] Jeff Heflin. Owl web ontology language-use cases and requirements. *W3C Recommendation*, 10:12, 2004.

[3] Neo4j. cypher-query-language.

[4] Roel Popping. Knowledge graphs and network text analysis, 2003.

[5] Evren Sirin and Bijan Parsia. Sparql-dl: Sparql query for owl-dl. In *OWLED*, volume 258, page 2, 2007.

[6] Lei Zhang. *Knowledge graph theory and structural parsing*. Twente University Press, 2002.