Analyse von Zeitreihen mit Linked Open Data

Analysis of Timeseries with Linked Open Data

Bachelor-Thesis von Simon Holthausen aus Frankfurt am Main

25.4.2013

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Knowledge Engineering

Analyse von Zeitreihen mit Linked Open Data
Analysis of Timeseries with Linked Open Data

Vorgelegte Bachelor-Thesis von Simon Holthausen aus Frankfurt am Main

1. Gutachten: Prof. Johannes Fürnkranz
2. Gutachten: Heiko Paulheim

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 25. April 2013

(Simon Holthausen)

Zusammenfassung

Das Semantic Web hat zum Ziel, Informationen maschinenlesbar zu machen, indem die Information nach festen Schemata hinterlegt werden. Diese festen Regeln und Strukturen ermöglichen viele neue, interessante Applikationen. In dieser Bachelorarbeit wird versucht, Analyse von Zeitreihen zu betreiben. Dafür werden die Vorteile des Semantic Webs und im Speziellen die Linked Open Data verwendet. Ziel ist es, passende Events zu einer Zeitreihe zu finden, die den Verlauf dieser Zeitreihe erklären. Ein Beispiel wäre der Aktienkurs einer Firma. Geht der Graph nach oben, sollen passende Nachrichten, die diesen Verlauf erklären, gefunden werden, zum Beispiel "Rekordgewinne vermeldet". In dieser Bachelorarbeit erforsche und evaluiere ich verschiedene Ansätze, um dieses Ziel zu erreichen.

Inhaltsverzeichnis

1 Einleitung und Motivation

In der heutigen Zeit ist es leichter denn je, über die Geschehnisse in der Welt informiert zu sein. Früher waren die maßgeblichen Nachrichtenorgane die Printmedien, das Radio und das Fernsehen. Das Radio und das Fernsehen senden immer nur zu bestimmten Uhrzeiten die Nachrichten, die Zeitung gibt es nur sechs Mal pro Woche. Durch das Internet ist es den Menschen nun möglich, überall und jederzeit an Nachrichten zu kommen. Sie haben außerdem eine sehr viel höhere Auswahl an Nachrichtenquellen. Das hat auch seine Nachteile, denn die Flut an Nachrichten muss durch den Nutzer gefiltert werden. Wenn er sich nur für ein bestimmtes Thema interessiert, werden viele Nachrichten dem Nutzer einfach überflüssig erscheinen. Eine Filterung der Nachrichten wird dem Nutzer nur teilweise abgenommen, etwa indem er auf Nachrichtenseiten auf den Sportbereich klickt und so nur sportrelevante Nachrichten sieht, oder indem er in Google entsprechende Suchtherme eingibt und sich die zugehörigen Nachrichten anzeigen lässt.

Noch schwieriger wird es für den Nutzer, wenn ihn nicht nur das Was und Wie, sondern das Warum interessiert. Wenn beispielsweise die Umfragewerte der Piratenpartei abstürzen, so berichten zahlreiche Nachrichtenseiten darüber. Nur die wenigsten gehen aber der Frage nach, warum das eigentlich passiert ist. Es könnte zum Beispiel auch interessant sein, warum der Aktienkurs eines Unternehmens im letzten Jahr erst sehr anstieg und dann fiel. Auch hier liefern die wenigsten Seiten einen Überblick, warum dies so geschehen ist. Der Nutzer ist auf sich gestellt und muss sich die Antworten selbst zusammensuchen. Zum Beispiel, indem er alle Nachrichten zu besagtem Unternehmen zusammensucht und sich bemüht herauszufinden, welche Nachrichten den Kursverlauf erklären könnten. Die Nachricht ”Rekordgewinne vermeldet” würde zum Beispiel den Anstieg des Kurses erklären, ”Die Wirtschaftskrise erfasst immer mehr Firmen” den Fall des Kurses.

Vor allem für Graphenverläufe ist die Frage nach dem Warum also selten schnell geklärt. Dabei ist diese Frage vor allem im Wirtschaftsbereich von großer Bedeutung. Aktienhändler sind sehr daran interessiert, die Graphenverläufe erklären zu können und damit Schlüsse für die Zukunft ziehen zu können. Aber auch für andere Bereiche kann die Erklärung eines Graphenverlaufs sehr wichtig sein. An dieser Stelle setzt diese Bachelorarbeit an. Es wird versucht, automatisch Erklärungen für Verläufe von Graphen zu finden. Das kann für Laien nützlich sein, die eine schnelle und kurze Erklärung haben möchten. Auch Experten können davon profitieren, indem sie durch eine automatische Zusammenstellung von Nachrichten einen ersten Überblick für eine umfassendere und tiefgründigere Analyse erhalten.

2 Grundlagen

Eine große Herausforderung im Zeitalter des Internets ist es, die vorhandenen Daten maschinell zu verarbeiten. Ein (noch) utopisches Ziel ist es zum Beispiel, in eine Suchmaschine einfach ”Ich will Urlaub auf Hawaii machen” einzugeben, und die Suchmaschine liefert daraufhin automatisch passende Urlaubszeiten, verlinkt Hotels in passenden Preisklassen und die zugehörigen Flüge. Das heißt, die Informationen, die der Mensch sich heute mühsam einzeln zusammensuchen muss, werden automatisch erfasst, verarbeitet, kombiniert und dem Nutzer präsentiert. Um dieses Ziel zu erreichen, existieren momentan zwei konträre Ansätze. Die Disziplin Web Mining versucht, die unstrukturierten und uneinheitlichen Daten aus Webseiten und Datenbanken mit Hilfe von Prozessen zu verarbeiten und in ein maschinenlesbares Format umzuwandeln. Der Ansatz des Semantic Web geht hier den umgekehrten Weg: Das Ziel ist es, die Daten direkt in standardisierten und strukturierten Formen zu präsentieren. Der Prozess des Umwandelns fällt also weg, an seine Stelle tritt der Prozess, alle Daten maschinenlesbar zur Verfügung zu stellen und Standards zu schaffen. Für die Analyse der Zeitreihen machen wir uns den zweiten Ansatz, den des Semantic Web zunutze.

2.1 Semantic Web

Das Konzept des Semantic Webs geht auf den Vorschlag von Tim Berners-Lee, Jim Hendler und Ora Lassila zurück [1]. Sie beschrieben 2001 in einer Ausgabe des Scientific American die Grundrisse des Semantic Webs. Der Kern des Semantic Web ist es, die Informationen in maschinenlesbarer Form bereitzustellen.

Der dabei am weitesten verbreitete Standard, der auch in dieser Arbeit verwendet wird, ist RDF. RDF steht für Resource Description Framework und ist ein Standard des W3C [4]. Das W3C (World Wide Web Consortium) ist eine internationale Vereinigung, die offene Standards entwickelt, um das geordnete Wachstum des Webs auf lange Sicht sicherzustellen [5]. Ein RDF-Dokument ist im Wesentlichen nichts anderes als ein gerichteter Graph, wobei die Kanten und Knoten eindeutige Namen besitzen. RDF-Dokumente können problemlos zusammengeführt werden. Das macht sie flexibel und für das Web geeignet.

Die Informationen, die RDF enthält, werden in Tripeln ausgedrückt. Ein Tripel besteht aus Subjekt, Prädikat und Objekt. Im Graphen wäre das eine Verbindung von einem Knoten über eine Kante zu einem anderen Knoten. Ein Beispiel ist <Darmstadt,stadtIn,Hessen> oder <TU Darmstadt,istEine,Universität>. Das Subjekt und das Prädikat müssen hierbei immer auflösbare Uniform Resource Identifier, kurz URIs sein, also an eine feste Ressource im Internet gebunden sein [9]. Der Sinn ist, das jedes Subjekt und Prädikat eindeutig über die URI identifizierbar und referenzierbar ist. Man kann also, wenn man möchte, der URI folgen und von ihr aus weitere Informationstripel verfolgen. Das Objekt kann ebenfalls eine Ressource sein, es kann sich aber auch um ein Literal handeln. Ein Literal kann keine weiteren Eigenschaften haben, es ist also ein einfacher String oder eine Zahl, die nicht als URI aufgelöst wird. Die Prädikate sollen nach Möglichkeit Standards folgen. Hierzu wurden im Laufe der Zeit verschiedene Schemata entworfen. Ein Beispiel ist foaf (Friend of a Friend) [10]. Prädikate hier sind zum Beispiel foaf:name oder foaf:knows. Ein Anwendungsfall wäre <Donald,foaf:knows,Dagobert>. Ressourcen sind Klassen und können so voneinander abhängen oder Subklassen voneinander sein. Ein Beispiel wäre <Auto,subKlasseVon,Fahrzeug>. Wie man sieht, ist also auch die Subklassenbeziehung ein Tripel der Form <Subjekt,Prädikat,Objekt>.

Dadurch, dass Ressourcen auflösbare URIs sind und mit Hilfe von Prädikaten auf andere Ressourcen verweisen können, ergibt sich ein Netz von Verlinkungen zwischen Ressourcen. Dass jede Ressource so viele und so akkurate Informationen wie möglich auf sich vereint und dabei sinnvoll auf andere Ressourcen verlinkt, das ist das Ziel der Linked Open Data.

2.2 Linked Open Data

Die Definition von Linked Open Data findet sich schon im Namen: aufeinander verlinkte, frei zugängliche Daten. Daten können dabei alles sein: Informationen über den US-Präsident, die Stadt Darmstadt oder auch eine kleine Informationssammlung über sich selbst. Ziel ist es, Informationen, die miteinander in Verbindung stehen, auch miteinander zu verlinken. Eine Informationskette könnte dann zum Beispiel so aussehen: <BarackObama,istEin,US-Präsident>, <US-Präsident,präsidentVon,Amerika>,<Amerika,entdecktVon,ChristopherColumbus>. Tim Berners-Lee hat für die Linked Open Data vier Grundprinzipien [3] aufgestellt:

1. ”Use URIs as names for things”
2. ”Use HTTP URIs so that people can look up those names.”

3. ”When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)”
4. ”Include links to other URIs. so that they can discover more things.”

Mit diesen vier Grundprinzipien soll sichergestellt werden, dass die Daten in einheitlicher Form und miteinander vernetzt vorzufinden sind. Dies in Verbindung mit der freien Zugänglichkeit und der Möglichkeit, dass jeder neue Informationen hinzufügen kann, machen Linked Open Data zu einer sehr mächtigen Informationsquelle.

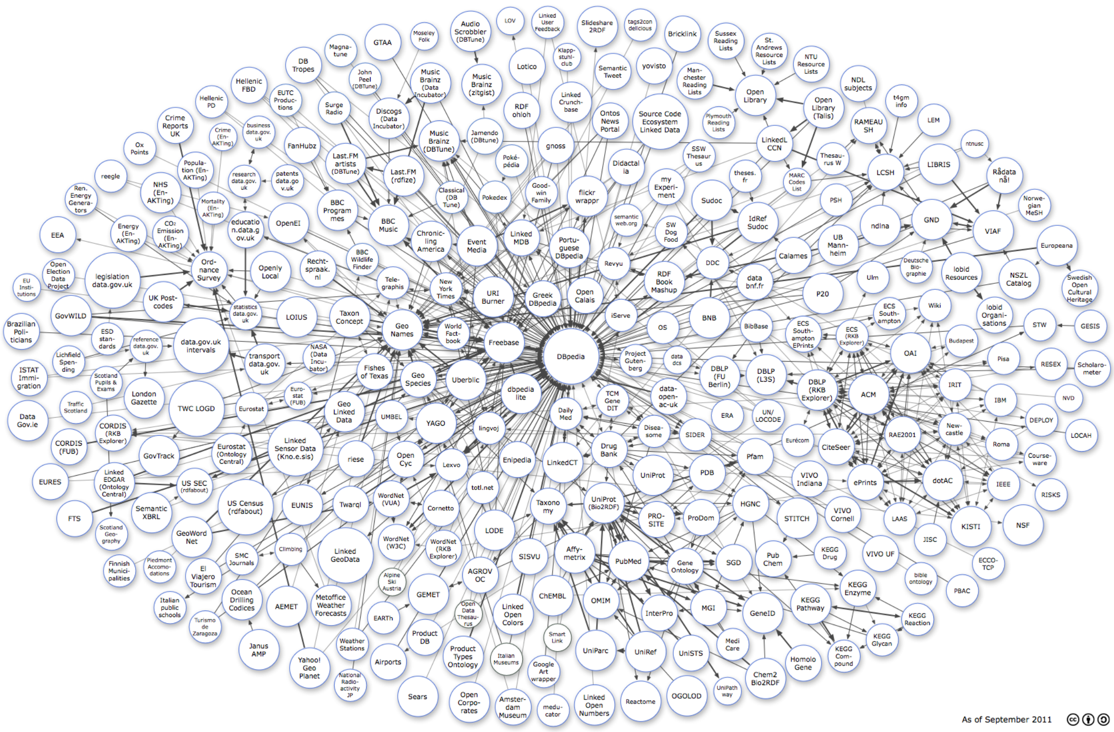

Abbildung 2.1: Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

Wo findet man nun die Linked Open Data? In der Linked Open Data Cloud. Hierbei handelt es sich um einen Verbund von Datenbanken, Schemata und allgemeinen Definitionen. Das ”Herz” der Cloud und die vielleicht bekannteste Datenbank aus dem Semantic Web ist die dbpedia. Der Name ist absichtlich verwandt mit der wikipedia. Den Grundstock an Daten, den die dbpedia umfasst, wurde aus den Infoboxen der Wikipedia generiert. Mittlerweile enthält die dbpedia zu vielen Ressourcen auch Abstracts in verschiedenen Sprachen, externe Weblinks und mehr. Insgesamt enthält die dbpedia über 3,7 Millionen Entitäten mit mehr als 1,8 Milliarden Tripeln und Verweise auf mehr als 100 weitere Datensets [2]. Die dbpedia ist nicht nur für Maschinen zugänglich, man kann sie auch ganz normal im Internet browsen.

dbpedia-owl:binomialAuthority	dbpedia:Moritz_Balthasar_Borkhausen
dbpedia-owl:class	dbpedia:Eudicots
dbpedia-owl:division	dbpedia:Flowering_plant dbpedia:Angiosperms
dbpedia-owl:family	dbpedia:Maloideae dbpedia:Rosaceae dbpedia:Spiraeoideae
dbpedia-owl:genus	dbpedia:Malus
dbpedia-owl:kingdom	dbpedia:Plant
dbpedia-owl:order	dbpedia:Rosales dbpedia:Rosids
dbpedia-owl:synonym	Malus pumila auct. Malus communis Desf. Pyrus malus L.
dbpedia-owl:thumbnail	http://upload.wikimedia.org/wikipedia/commons/thumb/1/15/Red_Apple.jpg/200px-Red_Apple.jpg
dbpedia-owl:wikiPageExternalLink	http://faostat.fao.org/site/339/default.aspx https://www.thieme-connect.com/ejournals/pdf/plantamedica/doi/10.1055/s-0028-1088300.pdf http://www.nationalfruitcollection.org.uk/ http://www.brogdalecollections.co.uk/ http://advances.nutrition.org/content/2/5/408.full.pdf+html http://www.ifr.ac.uk/info/society/spotlight/apples.htm http://gvsu.cdmhost.com/cdm4/results.php?CISOOP1=all&CISOBOX1=&CISOFIELD1=CISOSEARCHALL&CISOOP2=exact&CISOBOX2
dbpprop:b	Apples
dbpprop:binomial	Malus domestica
dbpprop:binomialAuthority	Borkh., 1803
dbpprop:calciumMg	6 (xsd:integer)
dbpprop:carbs	13.81
dbpprop:colwidth	30 (xsd:integer)

Abbildung 2.2: Ausschnitt der Browseransicht der dbpedia

Die dbpedia ist auch ein wichtiger Bestandteil dieser Arbeit. Sie dient als Ausgangspunkt, um verwandte Ressourcen zu einer gegebenen Ressource zu finden.

SPARQL

Wir Nutzer können uns über den Browser durch die Linked Open Data klicken. Die Maschine kann das in dieser Form nicht. Trotzdem muss es ihr möglich sein, auf die Informationen zugreifen zu können - oder mehr noch, sie kombinieren zu können, um daraus neue Erkenntnisse zu gewinnen. Hierfür gibt es die Abfragesprache SPARQL, die ähnlich SQL ist. SPARQL ist wie auch RDF ein Standard des W3C [4].

SPARQL stellt die Abfragesprache an die Datensets der Linked Open Data Cloud dar. Jedes Datenset besitzt einen SPARQL-Endpoint, über den mit Queries Abfragen ausgeführt werden können. Der Computer kommuniziert über diese Endpoints mit den Datensets und erhält Ergebnisse auf seine Anfragen. Eine simple Query, die alle Ressourcen zurückliefert, die das Prädikat label besitzen, sieht so aus:

```
SELECT ?x WHERE { ?x label ?y }
```

Die WHERE-Klausel ist ein zentraler Bestandteil von SPARQL. Vor dem WHERE stehen Variablen. Variablen werden durch ein vorangestelltes Fragezeichen gekennzeichnet. An die Variablen vor dem WHERE werden später die Ergebnisse gebunden. Diese Ergebnisse erfüllen die Bedingungen, die innerhalb der geschweiften Klammern formuliert sind. Im obigen Beispiel werden also alle ?x zurückgeliefert, bei denen gilt <?x,label,?y>. Es gibt weitere Sprachelemente, die die Abfrage verfeinern können. FILTER zum Beispiel filtert aus den vorhandenen Ergebnissen nochmals aus. Beispiel:

```
SELECT ?x WHERE { ?x label ?y} FILTER( regex(?y,"obama"))
```

In diesem Beispiel werden nur Ergebnisse zurückgeliefert, bei denen das label auch den String obama enthält. regex(?variable,string) ist hierbei eine von vielen vorhanden Methoden, die innerhalb von FILTER verwendet werden können. Weitere Operatoren sind zum Beispiel =, >, < um auf Gleichheit oder im Fall von Zahlen auch auf ”größer als”, ”kleiner als” zu prüfen.

3 Bestehende Ansätze zur Zeitreihenanalyse

Im Finanzsektor ist die Analyse von Zeitreihen ein Kernpunkt, um Voraussagen über Kursentwicklungen treffen zu können. Die Ansätze zielen allerdings weniger darauf ab, vergangene Kursverläufe umfassend zu erklären. Vielmehr geht es darum, aus dem Kursverlauf der Vergangenheit eine möglichst akkurate Vorhersage für die Zukunft zu treffen. Klassische Analysetechniken beruhen daher vor allem auf Mathematik und Taktik.

Eine auf den ersten Blick simple Technik ist die des Trends [11]. Der Graphverlauf wird über einen bestimmten Zeitraum angeschaut und analysiert, ob sich ein Trend ablesen lässt, daher ob sich der Graph überwiegend in eine bestimmte Richtung bewegt. Dazu werden Trendlinien an den lokalen Extrema des Graphen eingezeichnet, die von den Kursschwankungen abstrahieren und durch einen linearen Trend ersetzen. Genauer wird es mit Trendkanälen. Diese geben an, in welchem Bereich man die zukünftige Kursentwicklung erwarten kann. Sie geben also eine Unter- und Obergrenze an. Eine weitere Technik ist die Ermittlung des Durchschnitts. Dadurch werden Kursschwankungen gefiltert und es kann auf dieser Grundlage eine Vorhersage für die Zukunft getroffen werden.

Dazu kommt die Taktik [13]. Ein Börsenhändler kann sich zum Beispiel das Ziel setzen, auf jeden Fall ab einem bestimmten Punkt seine Aktie zu verkaufen - wenn sie einen bestimmten Wert überschreitet, um sicheren Gewinn einzufahren, oder wenn sie einen bestimmten Wert unterschreitet, um Verluste gering zu halten. Auch dazu bedient man sich der Analyse des Graphen, nimmt die Vorhersagen des Kursverlaufs zur Grundlage. In die Vorhersage können jetzt nicht nur mathematische Modelle, sondern auch Insiderinformationen einfließen. Wie man sieht, sind diese Techniken aber weniger dazu geeignet, um herauszufinden, warum der Graph einen bestimmten Verlauf genommen hat. Vor allem im Finanzsektor ist man hier mehr an der Zukunft interessiert.

Die klassische Methode, einen Graphen zu analyiseren, ist von Hand durch Experten. Vor allem in der Politik ist diese Methode vorherrschend. Im Wahlkampf stellen Parteien viele Analysten ein, um herauszufinden, wie sie mehr Stimmen auf sich vereinen können oder wo ihre Schwachstellen liegen. Neutrale Beobachter und Nachrichtenseiten kommentieren ihrerseits und stellen ebenfalls Analysen auf. Im US-Wahlkampf zwischen Obama und Romney zum Beispiel sahen viele Beobachter Obama vorne, da er viele Minderheiten für sich einnehmen konnte. Romney verlor die Wahl, da er vor allem als Kämpfer für die Reichen und Konzerne wahrgenommen wurde [12]. Diese Form der Analyse geht allerdings kaum auf den Graphverlauf im Einzelnen ein, sondern betrachtet den Graph in der jüngeren Zeit und zieht ein abschließendes Fazit. Das macht insofern Sinn, als sich Wahlkämpfer oder Zuschauer von Nachrichtenmagazinen kaum dafür interessieren, warum Obama vor einem Jahr noch bessere Umfragewerte erzielte. Ihnen geht es vorrangig um die jüngste Vergangenheit.

Eine Analyse eines Graphens über einen längeren Zeitraum gibt es also nur sehr selten und eine automatische Zusammenstellung von Nachrichten, die einen Graphverlauf erklären können, existiert noch nicht. Hier setzt die Bachelorarbeit an. Es wird versucht, mit Hilfe von Linked Open Data und einem Datenset, das Nachrichten aus unterschiedlichen Bereichen enthält, eine möglichst allgemeine Methode zur automatisierten Erklärung von Graphenverläufen zu finden.

4 Das Programm zur Zeitreihenanalyse

Das Programm für die Zeitreihenanalyse ist als Webapplikation realisiert. Verwendet werden HTM, CSS, Javascript, JSP, Servlets sowie Java. HTML, CSS und Javascript kommen auf Clientseite zum Einsatz. Die Eingaben des Nutzers werden an Servlets weitergeleitet. Diese wiederum leiten, je nach Aufgabe, an die entsprechenden Javaklassen weiter. Hier ist die Logik des Programms zu finden.

4.1 Aufbau des Programms

Das Programm ist folgendermaßen aufgebaut: Zunächst wählt der Nutzer eine CSV-Datei von seinem Computer aus. Die Datei muss dabei folgendem Format entsprechen:

```
XLabel;YLabel
Datum;Wert
...
```

wobei Datum in der Form Jahr-Monat-Tag vorliegen muss und der Wert eine beliebige Zahl sein kann. Ist die Zahl ein Kommawert, so muss sie in der Form VorkommaZahl,NachkommaZahl vorliegen. Die Datei wird eingelesen.

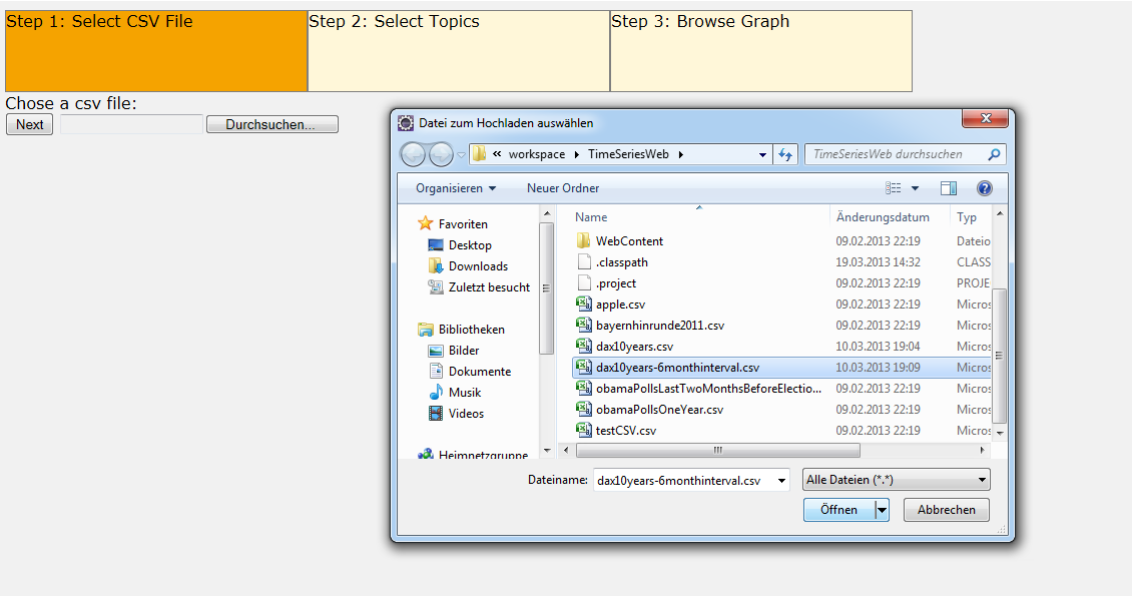

Abbildung 4.1: Auswählen einer CSV-Datei

Als Nächstes wählt der Nutzer die Themen aus, die mit diesem Graphen zu tun haben. Das geschieht über Autocomplete, wobei hier auf den LookupService von dbpedia zugegriffen wird. Der LookupService mit Prefixsearch sucht nach den am besten passenden Ergebnissen zur Eingabe. Auf die Eingabe `App` werden zum Beispiel `Apple`, `Apple_Inc.` und `Appleton,_Wisconsin` vorgeschlagen.

Über diese Suche kann der Nutzer beliebig viele Themen auswählen. Die Themen werden in einer Liste angezeigt, aus der man einzelne Einträge bei Bedarf auch wieder löschen kann. Ist man mit der Auwahl der Themen zufrieden, drückt man auf Start.

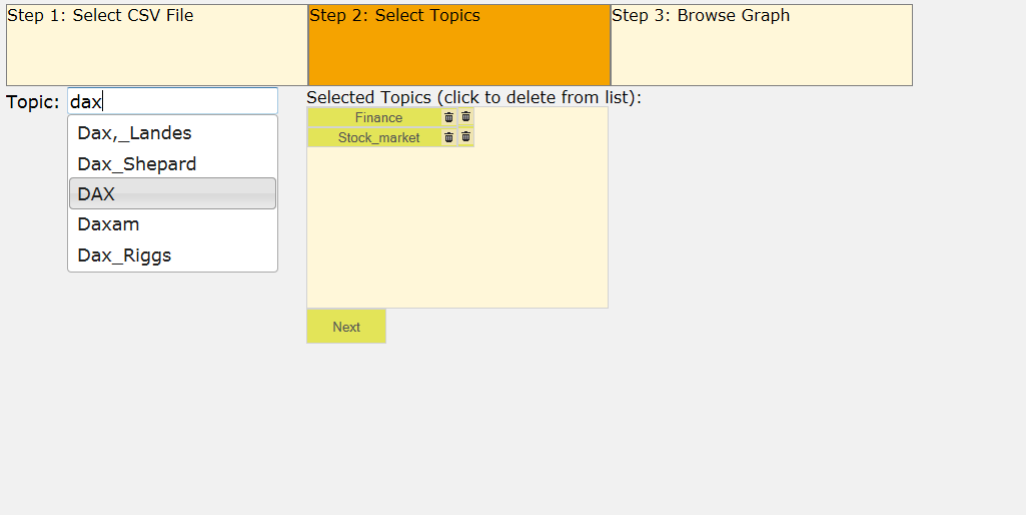

Abbildung 4.2: Themenauswahl

An dieser Stelle kommt der Kern des Programms zum Einsatz. Nach einem bestimmten Vorgehen wird aus der Liste der gegebenen Themen eine erweiterte Liste an Themen erzeugt, um so möglichst viele und passende Events zu finden.

Beispiel: Der Nutzer lädt eine CSV-Datei mit dem Apple-Aktienkurs von 2011 hoch. Er wählt als Thema `Apple_Inc.` aus. Aus dem Datenset werden also alle Events gefunden, die mit der Ressource `Apple_Inc.` in Verbindung stehen. Eine Meldung über den Tod von Steve Jobs etwa könnte nicht mit dieser Ressource verbunden sein, sondern nur mit der Ressource `Steve_Jobs` als Person. Da Apple und Steve Jobs aber eng in Beziehung zueinander stehen, nimmt das Programm bei der Suche nach Events auch die Ressource `Steve_Jobs` auf.

Die Verfahren, nach denen die Ressourcen zur Suche nach Events ausgewählt wurden, werden im Kapitel fünf ausführlich besprochen. Verschiedene Verfahren werden dort vorgestellt und evaluiert.

Ist die Suche nach weiteren Ressources durch das Programm abgeschlossen, sucht das Programm die passenden Events. Dazu werden zu je zwei Datenpunkten ein Zeitintervall gebildet in der Form [Datenpunkt a + 1 Minute, Datenpunkt b]. Nun wird für jedes Zeitintervall nach Events gesucht, die im momentan betrachteten Intervall liegen und mit einer der gesuchten Ressourcen verlinkt sind. Die Beschreibung der Events, also der String, der die News beschreibt, wird dann zurückgegeben. Der Nutzer kann sich nun den Graphen anschauen, auf die Datenpunkte klicken und sich so anzeigen lassen, was das Programm zu diesem Zeitpunkt gefunden hat. Klickt man auf Datenpunkt n, werden also alle Events angezeigt, die im Intervall [Zeitpunkt Datenpunkt n-1 + 1 Minute, Zeitpunkt Datenpunkt n] liegen. Die Events, die für einen Datenpunkt hinterlegt sind, erklären also immer, wie es zu dem Graphverlauf bis zu diesem Zeitpunkt kam.

Beispiel: Der Nutzer klickt im Graph auf den Datenpunkt zum Zeitpunkt 1.11.2011, 0Std0Min. Der Datenpunkt davor datiert auf den 1.10.2011, 0Std0Min. Der Nutzer sieht also alle passenden Events, die im Intervall [1.10.2011, 0Std1Min , 1.11.2011, 0Std0Min] liegen.

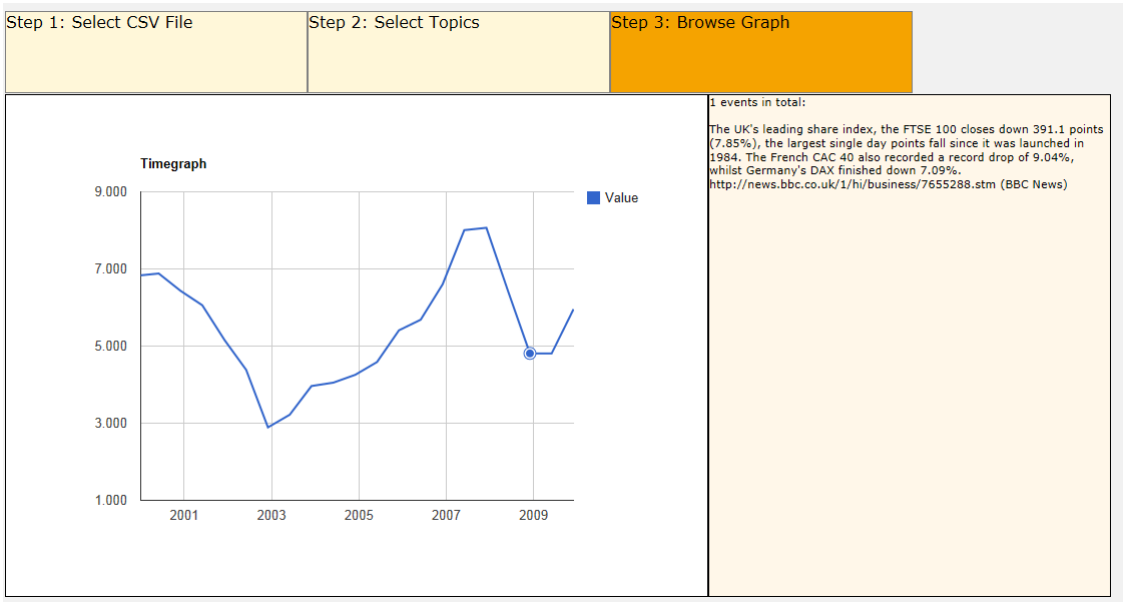

Abbildung 4.3: Graph browsen

Der Nutzer kann jederzeit einen Schritt zurück gehen, weitere oder andere Themen hinzufügen und erneut nach passenden Events suchen, oder einen anderen Graphen laden.

4.2 Das LOD Datenset

Um für die Zeitreihenanalyse passende Events zu finden, nutzt das Programm das LOD Datenset. Es wurde im Rahmen des Papers ”Automatic Classification and Relationship Extraction for Multi-Lingual and Multi-Granular Events from Wikipedia” von Daniel Hienert, Dennis Wegener und Heiko Paulheim erstellt und wird von GESIS weiterentwickelt [6]. Es wurden automatisch Nachrichten aus Wikipedia extrahiert und verarbeitet. Die Nachrichten wurden mit Ressourcen getaggt, von denen die Nachrichten handeln. Die Arbeiten basieren auf dem Paper ”Extraction of Historical Events from Wikipedia” von Daniel Hienert und Francesco Luciano [7]. Momentan steht nur eine lokale Version des Datensets zur Verfügung, Ziel ist es aber, in Zukunft online auf den SPARQL-Endpoint zugreifen zu können, um immer aktuelle Daten zu erhalten.

Die Struktur des Datensets ist vorgegeben und sehr simpel. Jedes Event liegt als Ressource vor, zu dem eine Reihe von zusätzlichen Informationen hinterlegt sind. Jedes Event enthält eine Beschreibung, Links auf Ressourcen, mit denen dieses Event zu tun hat, und einen Timestamp, der angibt, von wann dieses Event ist.

```
<http://lod.gesis.org/historicalevents/HE5140c5f839d669a8ee17058449f646a3>
dcterms:isPartOf <http://lod.gesis.org/historicalevents/>;
dcterms:description "President Slobodan Milošević leaves office after widespread demonstrations
    throughout Serbia and the withdrawal of Russian support. This political event became known as 5th
    October Revolution in Serbia."@en;
lode:involvedAgent <http://dbpedia.org/resource/Slobodan_Milo%C5%A1evi%C4%87>;
lode:involvedAgent <http://dbpedia.org/resource/Serbia>;
lode:involvedAgent <http://dbpedia.org/resource/5th_October_Overthrow>;
lode:atTime [a time:DateTimeInterval; time:xsdDateTime "2000-10-05T00:00:00Z"^^xsd:dateTime].
```

Abbildung 4.4: Beispiel für ein Event im LOD Datenset

Die hinterlegten Nachrichten haben hauptsächlich die Themen Politik, Wirtschaft und Weltgeschehen. Klatsch-und-Tratsch-Nachrichten sowie Sportereignisse finden kaum Beachtung. Das Datenset ist in zwei Teile gegliedert: Der erste Teil bietet ausgewählte Events aus den Jahren 300v.Chr. bis in das Jahr 2012. Insgesamt sind es mehr als 37800 Events. Der zweite Teil ist eine genauere Betrachtung der Jahre 2000 bis 2012. Hier gibt es insgesamt über 38000 Einträge. Das ergibt einen Schnitt von 263 Events pro Monat, wobei die Werte stark schwanken. So gibt es Monate mit über 1000 Einträgen, andere haben gerade einmal 10. Der Hauptteil der Events ist in den späteren Jahren zu finden.

5 Vorstellung der Methoden

In diesem Kapitel werden die Methoden vorgestellt, die benutzt werden, um ausgehend von den ursprünglich vom Nutzer eingegebenen Ressourcen weitere Ressourcen zu finden, die möglichst zum Thema passen. Ist zum Beispiel die Ressource `Apple_Inc` vom Nutzer gegeben, so wäre eine verwandte Ressource `IPhone`. Ob diese Ressource aufgenommen wird, hängt von der Methode ab. Die Methoden verfolgen unterschiedliche Ansätze, die im Folgenden erläutert werden.

Zur Suche nach neuen Ressourcen wird meistens die dbpedia genutzt. Dazu wird der SPARQL-Endpoint verwendet. An diesen werden SPARQL-Abfragen gestellt, die Ergebnisse stellen die neu gewonnenen Ressourcen dar.

5.1 Simple Methode: Nur gegebene Ressourcen

Bei dieser Methode werden nur die Ressourcen zur Suche nach Events verwendet, die vom Nutzer eingegeben wurden. Hat der Nutzer zum Beispiel einen Apple Aktienkurs Graph und gibt als Thema nur die Ressource `Apple_Inc.` ein, so wird das Programm auch nur Events suchen, die mit `Apple_Inc.` getaggt sind.

Dies ist die einfachste aller Methoden. Man geht hier davon aus, dass im Datenset alle Events umfassend und gut getaggt sind. So sollte jedes Event, das in irgendeiner Form mit der gegebenen Ressource zu tun hat, diese Ressource als involvedAgent haben. Gleichzeitig wird davon ausgegangen, dass Events, die nicht direkt mit den gegebenen Ressourcen zu tun haben, nicht relevant sind und den Graphen nicht erklären können. Eine Nachricht über die allgemeine positive Lage am Aktienmarkt wird also nicht als relavant für den Apple-Aktiengraphen im Speziellen angesehen. Das könnte unter Umständen ein Fehler sein und dazu führen, das Events verpasst werden, die relevant sind, auf der anderen Seite verspricht man sich davon eine sehr hohe Precision.

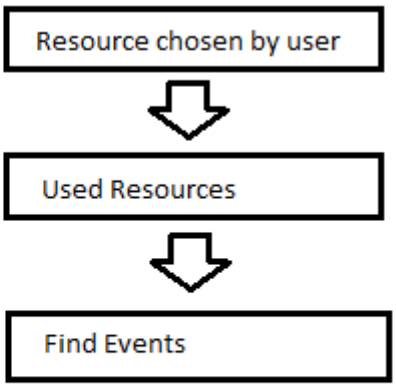

Abbildung 5.1: Simple Methode grafisch veranschaulicht

5.2 InOut Methode

Bei dieser Methode werden neben den vom Nutzer eingegebenen Ressourcen auch alle Ressourcen verwendet, die mit diesen verbunden sind. Das sind alle eingehenden und ausgehenden Links. Die Ressource `Bruce_Willis` ist zum Beispiel über das Prädikat `birthPlace` mit der Ressource `West_Germany` verbunden. Dies ist eine Ressource über einen ausgehenden Link, also <Bruce_Willis,birthPlace,West_Germany>. `Die_Hard` ist außerdem über das Prädikat `starring` mit `Bruce_Willis` verbunden. Diese ist eine Ressource über einen eingehenden Link, also <Die_Hard,starring,Bruce_Willis>. Das heißt dass bei dieser Methode neben der ursprünglichen Ressource `Bruce_Willis` auch `Die_Hard` und `West_Germany` bei der Suche nach Events berücksichtigt werden.

Die Idee hinter dieser Methode ist, das alle Ressourcen, die über In- und Outlinks mit den ursprünglichen Ressourcen verbunden sind, auch mit diesen zu tun haben und daher ebenfalls relevant bei der Suche nach passenden Events sind. Im obigen Beispiel von <Die_Hard,starring,Bruce_Willis> ergibt dies durchaus Sinn, denn Bruce Willis hat in diesem Film mitgespielt und Nachrichten zu Stirb Langsam werden mit hoher Wahrscheinlichkeit auch Informationen zu Bruce Willis beinhalten. Allerdings besteht dadurch die Gefahr, dass auch Ressourcen in die Suche mit einbezogen werden, die zu allgemein sind oder bei der Analyse eines Graphen nicht zum Thema passen. `Barack_Obama` zum Beispiel hat einen Link auf `United_States`. Das könnte wertvolle Nachrichten zu Umfragewerten während der Präsidentschaftswahl liefern, aber auch andere Nachrichten, die nicht zu diesem Thema passen. Generell verspreche ich mir von diesem Ansatz aber mehr gefundene Events, von denen viele auch zum Thema passen.

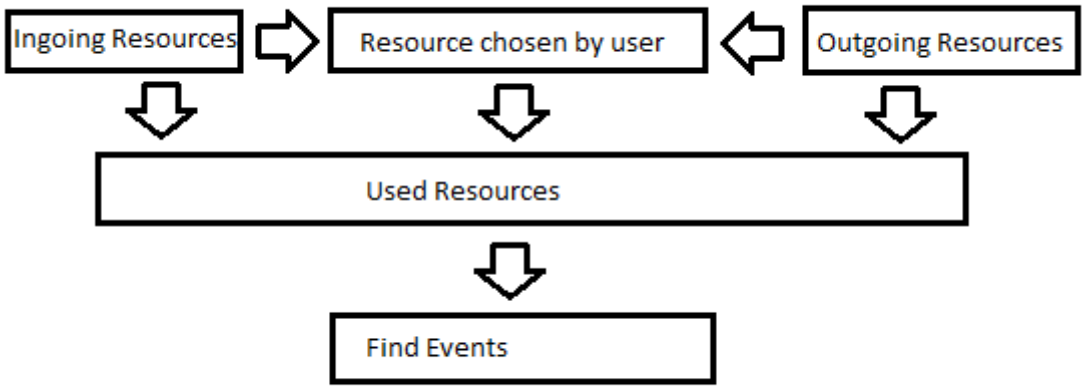

Abbildung 5.2: InOut Methode grafisch veranschaulicht

Die zugehörige SPARQL-Query an die dbpedia sieht folgendermaßen aus:

```
SELECT ?x WHERE {{ <givenResource> ?y ?x. } UNION {?x ?y <givenResource>}}
```

5.3 Ressourcen aus gefundenen Events Methode

Bei dieser Methode wird zunächst mit den vom Nutzer eingegebenen Ressourcen nach Events gesucht. Jedes Event hat gewöhnlich mehrere Ressourcen, mit denen es getaggt ist. Diese Ressourcen werden ausgelesen und in eine Liste aufgenommen. Diese so gefundenen Ressourcen werden nun genutzt, um nach weiteren Events zu suchen. Beispiel: Der Nutzer hat einen Obama Umfragegraph. Er gibt also die Ressource `Barack_Obama` ein. Eines der über diese Ressource gefundenen Events hat als weitere verlinkte Ressource `United_States`. Beim zweiten Durchlauf zur Suche nach weiteren Events werden also auch Events gesucht, die mit der Ressource `United_States` getaggt sind. Diese Methode ist die einzige, die nicht den SPARQL-Endpoint der dbpedia benutzt, um weitere Ressourcen zu finden.

Die Idee hinter diesem Ansatz ist, dass eine Nachricht nie von zu unterschiedlichen Themen handelt. Eine Nachricht von einem Treffen zwischen Nicolas Sarkozy und Barack Obama wird zum Beispiel mit den Ressourcen der beiden Politiker getaggt sein. Beide sind Politiker und in dieser Hinsicht themenverwandt. Dadurch, dass alle getaggten Ressourcen der gefundenen Events aufgenommen werden zur Suche nach weiteren Events, erhoffe ich mir, dass so Events gefunden werden, die ebenfalls zum Thema passen. Durch diese Methode können auch Ressourcen zur Suche aufgenommen werden, die nicht direkt ersichtlich sind und so neue interessante Erkenntnisse liefern könnten. Die Gefahr ist, dass das Themenspektrum sehr auseinandergehen kann. Sucht man zum Beispiel nach Events mit Barack Obama und findet Nachrichten zu Zypern-Gesprächen oder Europa-Treffen, so können die Ressourcen `Zypern` und `Europa` in die Liste an Ressourcen zur Suche aufgenommen werden, was zwar sehr viele neue Events liefert, viele davon aber wohl nichts mehr mit Barack Obama zu tun haben.

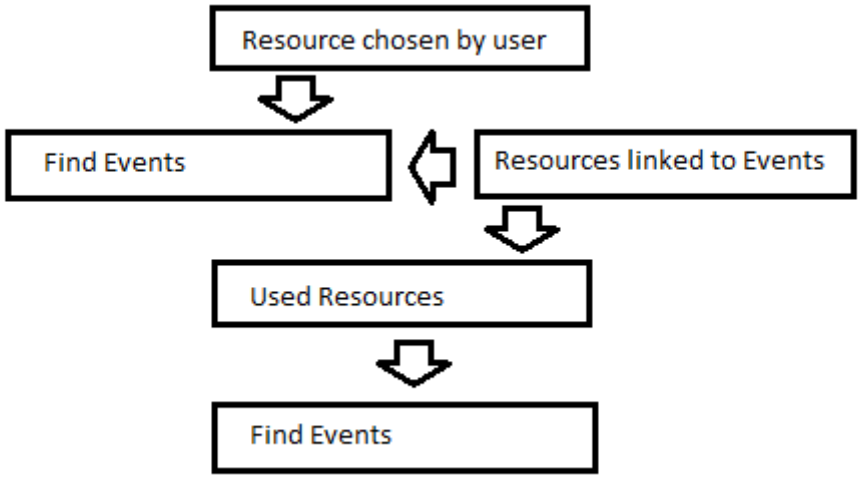

Abbildung 5.3: Ressourcen aus gefundenen Events Methode grafisch veranschaulicht

5.4 InOut Methode - Order by Relevance

Wie auch bei der *InOut Methode* werden neben den vom Nutzer eingegebenen Ressourcen auch alle Ressourcen verwendet, die mit diesen verbunden sind. Das sind alle eingehenden und ausgehenden Links. Alle so gefundenen Ressourcen werden allerdings nach ihrer Relevanz geordnet. Die Relevanz wird hierbei über das Prädikat dcterms:subject mal der Anzahl an Links zu einer Ressource gemessen.

Fast jede Ressource besitzt das Prädikat `dcterms:subject`. Dieses Prädikat gibt an, in welchem Themenbereich die gegebene Ressource angeordnet ist. Für die Ressource `IPhone_4S` wäre das zum Beispiel unter anderem `Touchscreen_mobile_phones`, `IPhone` und `Apple_Inc`. Diese Information wird folgendermaßen genutzt: Alle eingehenden und ausgehenden Ressourcen der ursprünglichen Ressourcen werden danach geordnet, wie viele Themen sie mit diesen gemeinsam haben.

Als Zweites wird berücksichtigt, wie viele verschiedene Links es auf die ein- und ausgehenden Links von der betrachteten Ressource aus gibt, also über wie viele Prädikate sie miteinander verbunden sind. Betracht man zum Beispiel <Barack_Obama,staatsbürgerVon,United_States> und <Barack_Obama,präsidentVon,United_States>, so gibt es zwei Links auf die Ressource `United_States` von der betrachteten Ressource `Barack_Obama` aus.

Diese beiden Maße werden nun miteinander multipliziert. Als endgültiges Relevanzmaß ergibt sich also *Anzahl gemeinsamer Themenbereiche von RessourceX und ursprünglicher Ressource* mal *Anzahl von Links zwischen ursprünglicher Ressource und RessourceX*.

Die Idee hinter diesem Ansatz ist, dass InOut-Links nicht unbedingt mit der ursprünglichen Ressource themenverwandt sein müssen. So verweist zum Beispiel `Apple_Inc` über das Prädikat `locatedIn` auf die Ressource `United_States`. Diese beiden Ressourcen haben keine direkte Gemeinsamkeit, außer dass A in B seinen Sitz hat. Diese Art InOut-Links helfen oft nicht weiter, da sie zu allgemein sind oder nicht zum Thema passen. Aus diesem Grund wird das Gemeinsamkeitsmaß über das Prädikat `dcterms:subject` bestimmt. Ressourcen, die im selben Themenbereich angesiedelt sind und gleichzeitig über einen Link mit der ursprünglichen Ressource verbunden sind, haben mit sehr hoher Wahrscheinlichkeit mit dem Thema zu tun. Potentiell unnütze Ressourcen der reinen InOut-Methode werden so also herausgefiltert, wodurch ich mir eine höhere Precision verspreche.

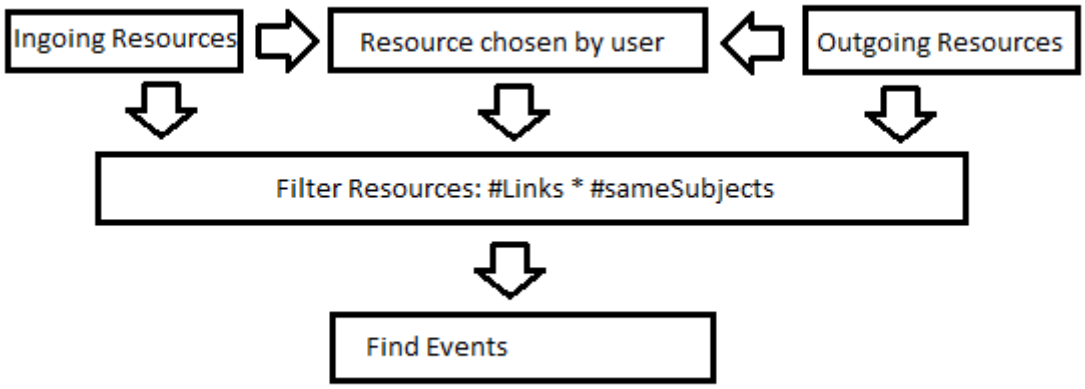

Abbildung 5.4: InOut Methode - Order by Relevance grafisch veranschaulicht

Die zugehörige SPARQL-Query an die dbpedia sieht folgendermaßen aus:

```
SELECT ?x (COUNT(?s) AS ?c0) WHERE {
                {SELECT ?x WHERE {{ <givenResource> ?y ?x. } UNION {?x ?y <givenResource>}}}
                ?x <http://purl.org/dc/terms/subject> ?s .
                <givenResource> <http://purl.org/dc/terms/subject> ?s}
ORDER BY DESC(?c0)
```

5.5 Über zwei Links verwandte Ressourcen - Order by Relevance

Bei dieser Methode werden zunächst alle vom Nutzer ursprünglich verwendeten Ressourcen betrachtet. Für jede dieser Ressourcen werden alle Ressourcen gesucht, die über zwei Links mit dieser verbunden sind. Zum Beispiel wäre `Barack_Obama` mit `United_States` über zwei Links verbunden, da <BarackObama,istEin,US-Präsident>, <US-Präsident,präsidentVon,United_States>. Da auf diese Weise sehr viele Ressourcen gefunden werden, ist eine Ordnung nach Relevanz zwingend, um am Schluss nur die ersten 60 bis 100 Ressourcen zu verwenden. Die Relevanz wird hierbei über die Anzahl der gemeinsamen Links über eine dritte Resource mal der Ähnlichkeit der beiden Resourcen in gemeinsamen Themenbereichen gemessen.

Die Ähnlichkeit hinsichtlich gemeinsamer Themenbereiche wird wie auch in *InOut Methode - Order by Relevance* gemessen. Das zweite Relevanzmaß berücksichtigt die Anzahl von Links über eine dritte Ressource. Beispiel: <RessourceA,prädikat1,RessourceX>, <RessourceX,prädikatx,RessourceC> sowie <RessourceA,prädikat2,RessourceB>, <RessourceY,prädikaty,RessourceC>. Das heißt, RessourceA und RessourceC sind zweimal über eine andere Ressource verlinkt. Sie haben also ein Relevanzmaß von zwei.

Dieses Relevanzmaß wird jetzt mit der Ähnlichkeit gemeinsamer Themenbereiche multipliziert. Hat eine Ressource also zwei gemeinsame Themenbereiche mit der ursprünglichen Ressource und ist dreimal über eine dritte Ressource mit der ursprünglichen Ressource verlinkt, so ergibt das eine Relevanz von 2*3=6.

Da auf diese Weise sehr viele Ressourcen gefunden werden, wird ein Limit eingeführt. Es werden 300 Ressourcen zurückgegeben - falls so viele gefunden werden, sonst weniger -, davon wird das erste Drittel genommen, mindestens jedoch 60. Für weniger Ressourcen ist die Gefahr hoch, dass nur sehr spezielle Ressourcen aufgenommen werden, für mehr Ressourcen werden einfach zu viele Events gefunden, die nicht mehr sinnvoll für die Erklärung von Graphen sind.

Die Idee aus dem *Order by Relevance*-Ansatz wird hier übernommen und weitergeführt. Unter Umständen können nur noch sehr wenige Ressourcen zur weiteren Suche übrig bleiben, wenn man nur die direkten InOut-Links der betrachteten Ressourcen miteinbezieht und dann nach gemeinsamen Themen filtert. Diese Themen sind außerdem oft spezieller als die ursprüngliche Ressource, so ist zum Beispiel die Ressource `IPhone` sehr viel spezieller als die ursprüngliche Ressource `Apple_Inc.` Um mehr Ressourcen für die weitere Suche zu finden, werden in diesem Ansatz also auch Ressourcen in Betracht bezogen, die über zwei Links mit der ursprünglichen Ressource verbunden sind. Dadurch erhoffe ich mir eine umfangreiche Liste an Ressourcen, die dennoch zum Thema passen, da sie entsprechend gefiltert wurden.

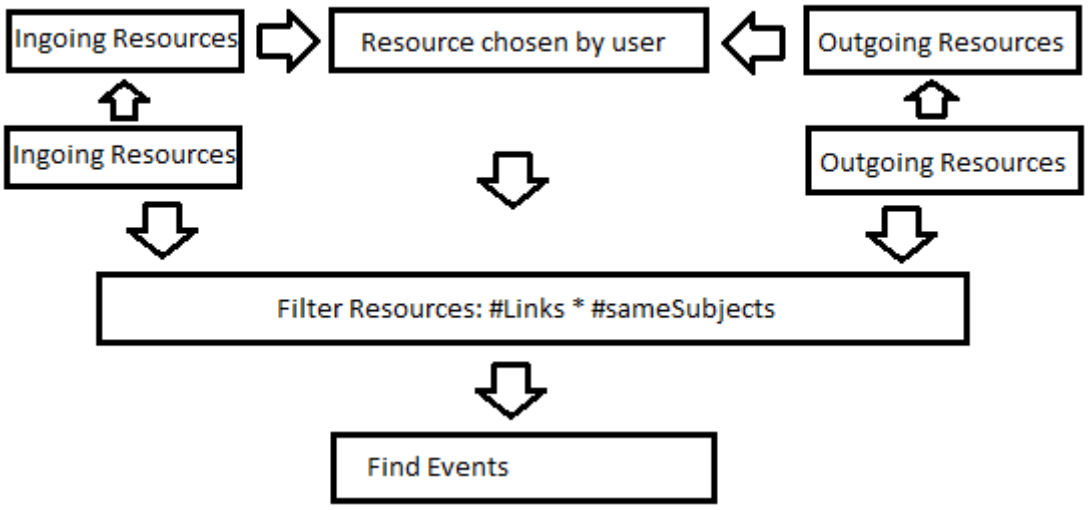

Abbildung 5.5: Über zwei Links verwandte Ressourcen - Order by Relevance grafisch veranschaulicht

Die zugehörige SPARQL-Query an die dbpedia sieht folgendermaßen aus:

```
SELECT ?x (COUNT(?s) AS ?c0) WHERE {
                {SELECT ?x WHERE {{<givenResource> ?y0 ?z0. ?z0 ?y1 ?x}
                        UNION {?x ?y0 ?z0. ?z0 ?y1 <givenResource>}}}
                ?x <http://purl.org/dc/terms/subject> ?s.
                <givenResource> <http://purl.org/dc/terms/subject> ?s}
ORDER BY DESC(?c0) LIMIT 300
```

5.6 Filter

Zusätzlich zu den Methoden gibt es Filter. Ein Filter wird auf alle gefundenen Events, die durch die Methoden gefunden wurden, angewendet und - wie der Name schon sagt - filtert noch einmal, um die Precision zu verbessern, möglichst ohne Events herauszufiltern, die wichtig sind.

5.6.1 Subject Broader Filter

Dieser Filter betrachtet alle gegebenen Events und filtert diejenigen heraus, die keine thematische Übereinstimmung mit den ursprünglichen Ressourcen hat. Eine thematische Übereinstimmung ist in folgenden Fällen gegeben:

Wie auch bei den *Order by Relevance*-Ansätzen der oben besprochenen Methoden wird die thematische Übereinstimmung über das Prädikat `dcterms:subject` bestimmt. Allerdings wird hier noch einen Schritt weiter gegangen. Die Ressourcen, auf die über `dcterms:subject` verlinkt wird, sind allesamt Kategorien. Diese verfügen über das Prädikat `broader`. Dieses Prädikat gibt an, dass die aktuell betrachtete Kategorie eine Unterkategorie der Kategorien ist, die dort gelistet sind. Umgekehrt gibt es auch Verweise auf Unterkategorien, also von welchen Kategorien die momentan betrachtete Kategorie eine Überkategorie ist. Dies geschieht mit dem umgekehrten Prädikat `broader of`. Diese `broader`- und `broader of`-Struktur wird ausgenutzt, um eine Liste von Kategorien zu erstellen, die zum Thema passen. Der Filter nimmt also zunächst alle ursprünglich vom Nutzer gegebenen Ressourcen und erstellt über diese Struktur eine Liste von Kategorien. Dabei werden alle Kategorien, die über einmal `broader`, und alle Kategorien, die über einmal, zweimal und dreimal `broader of` erreichbar sind, aufgenommen.

Als Beispiel sei die Ressource `Apple_Inc` gegeben. Über `dcterms:subject` ist unter anderem die Kategorie `Computer companies of the United States` verlinkt. Diese Kategorie hat als Überkategorie, also über das `broader`-Prädikat verlinkt, unter anderem die Kategorie `Computer hardware companies`. Diese und alle anderen direkten Überkategorien werden aufgenommen. Weiterhin ist `Computer companies of the United States` eine Überkategorie von `Defunct`

`computer companies of the United States`, also über `broader of` verlinkt. Diese und alle weiteren Unterkategorien werden in die Liste aufgenommen. Von diesen Unterkategorien wird jeweils noch einmal weiter nach unten gegangen, also auch deren Unterkategorien in die Liste aufgenommen. Für diese Unterkategorien wird der Schritt ein drittes und letztes Mal ausgeführt. Alle auf diese Weise gefundenen Kategorien stehen nun in einer Liste.

Nun wird jedes Event analyisiert und geprüft, ob eine der getaggten Ressourcen thematisch mit den Kategorien in der Liste zu tun haben. Dafür werden für die getaggten Ressourcen alle Kategorien des `dcterms:subject` Prädikats extrahiert. Findet sich eine dieser extrahierten Kategorien in der Liste der vorher zusammengestellten Kategorien, so weist das zugehörige Event eine thematische Übereinstimmung auf. Wird keine Übereinstimmung gefunden, so wird das Event herausgefiltert. Das Vorgehen dieses Filters wird in der Abbildung noch einmal veranschaulicht.

Die Idee hinter diesem Filter ist, dass Events thematisch in irgendeiner Form eine Übereinstimmung haben müssen, um überhaupt den Graphen erklären zu können. Die thematische Übereinstimmung wird hier über die Kategorien bestimmt. Dass nur eine Stufe von Überkategorien, aber drei Stufen von Unterkategorien genommen werden, ist sinnvoll. Mehr als eine Stufe von Überkategorien würde zu sehr allgemeinen Kategorien führen, die thematisch zu breit gefächert sind und für den speziellen Graphen daher zu weit gefasst sind. Auf der anderen Seite werden bis zu drei Stufen von Unterkategorien genommen, da hier nur eine Verfeinerung der Kategorien möglich ist. Auch die so gefundenen Unterkategorien sollten also das ursprünglich weiter gefasste Thema erklären können.

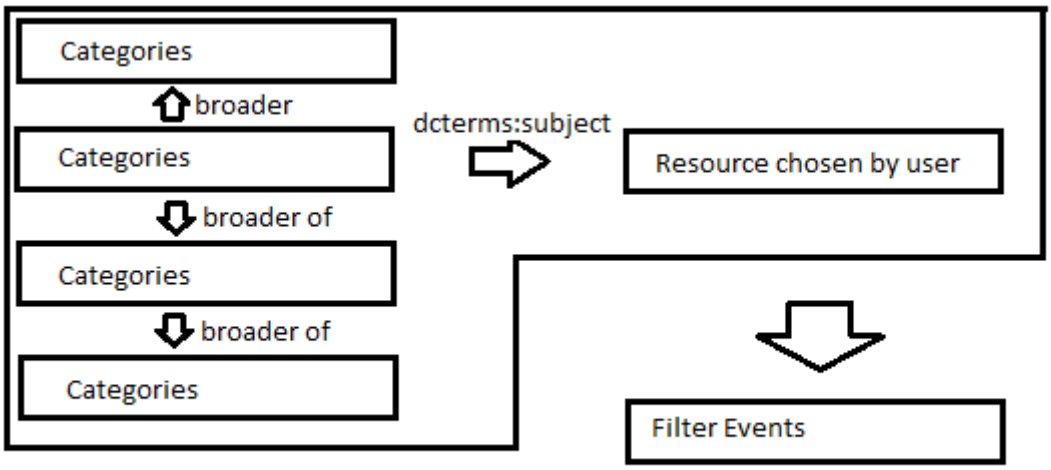

Abbildung 5.6: Subject Broader Filter grafisch veranschaulicht

5.6.2 UClassify Filter

Dieser Filter betrachtet alle gegebenen Events, und filtert diejenigen heraus, deren Aussage konträr zum Graphverlauf ist. Wenn also eine Nachricht sehr positiv ist, der Graphverlauf aber nach unten zeigt, so wird sie herausgefiltert. Heißt es zum Beispiel ”Apple vermeldet Rekordgewinn”, gleichzeitig sackt in diesem Zeitraum der Aktienkurs jedoch ab, so wird diese Nachricht herausgefiltert.

Um herauszufinden, ob eine Textaussage positiv, negativ oder neutral ist, wird der *SentimentClassifier* von *UClassify* verwendet [14]. Der Classifier wird über die Web-API, also mit einer URL mit den richtigen Parametern aufgerufen. Es wird ein html-Object zurückgeliefert, das zwei Werte enthält: den Positiv- und den Negativwert des Textes. Der Positivwert ist 1-Negativwert, zusammen ergeben sie 1. Ich habe die Werte folgendermaßen verarbeitet: Ist der Positiv- beziehungsweise Negativwert über 0,6, so wird der Text als positiv beziehungsweise negativ gewertet, sonst neutral. Sind die Werte über 0,8, so werden die Texte als sehr positiv oder sehr negativ gewertet. Die Nachrichten werden nach dieser Klassifizierung eingeteilt, also in *sehr positiv*, *positiv*, *neutral*, *negativ* und *sehr negativ*.

Weiterhin werden die Intervalle des Graphen betrachtet und diese in *Positiver Trend*, *Negativer Trend* und *Neutral* eingeteilt. Der Trend ist neutral, wenn sich der Graph nicht mehr als ein Zehntel im Verhältnis zum durchschnittlichen Wert verändert. Sind die Punkte des Graphen zum Beispiel bei 50, 140 und 70, so ist der durchschnittliche Wert $(90+70)/2=80$. Wenn die Veränderung des Graphen zwischen zwei Punkten nun unter einem Zehntel dieses Werts, also

8 liegt, dann wird der Trend hier als neutral angesehen. Steigt der Graph stärker an, so ist der Trend positiv, fällt der Graph stärker, so ist der Trend negativ.

Diese Klassifizierungen sind wichtig für den Filter: Ist der Graphverlauf postiv, so werden alle Events in der Klasse *sehr negativ* sofort herausgefiltert. Neutrale Events bleiben unangetastet. Dieses Vorgehen gilt umgekehrt für negative Graphverläufe. Ist der Graphverlauf neutral, so werden alle Events der Klasse *sehr negativ* und *sehr positiv* sofort herausgefiltert.

Die Idee ist hier, dass sehr negative Nachrichten bei einem positiven Graphverlauf nicht passen und den Verlauf nicht erklären können. Neutrale bis negative News können zum Verlauf passen, solange es auf der anderen Seite auch genügend positive News gibt. Das ergibt Sinn, denn wenn man zum Beispiel einen Aktienkursverlauf von Monat zu Monat betrachtet, so kann innerhalb eines Monats viel passieren, sowohl Positives als auch Negatives. Solange die positiven Nachrichten in der Überzahl sind, kann der Graphverlauf so erklärt werden.

6 Evaluation

In diesem Abschnitt evaluiere ich die getesteten Methoden zum Finden von Events. Als Grundlage für die Evaluation dienen vier Graphen: Ein Apple Aktienkurs Graph im Jahr 2012, ein Graph über Barack Obamas Umfragewerte vor der Präsidentschaftswahl 2012, ein Graph über den Punkteverlauf des FC Bayern aus der Hinrunde 2011 und ein DAX-Kursverlauf vom Jahr 2000 bis 2010. Im folgenden wird jeder Graph kurz vorgestellt.

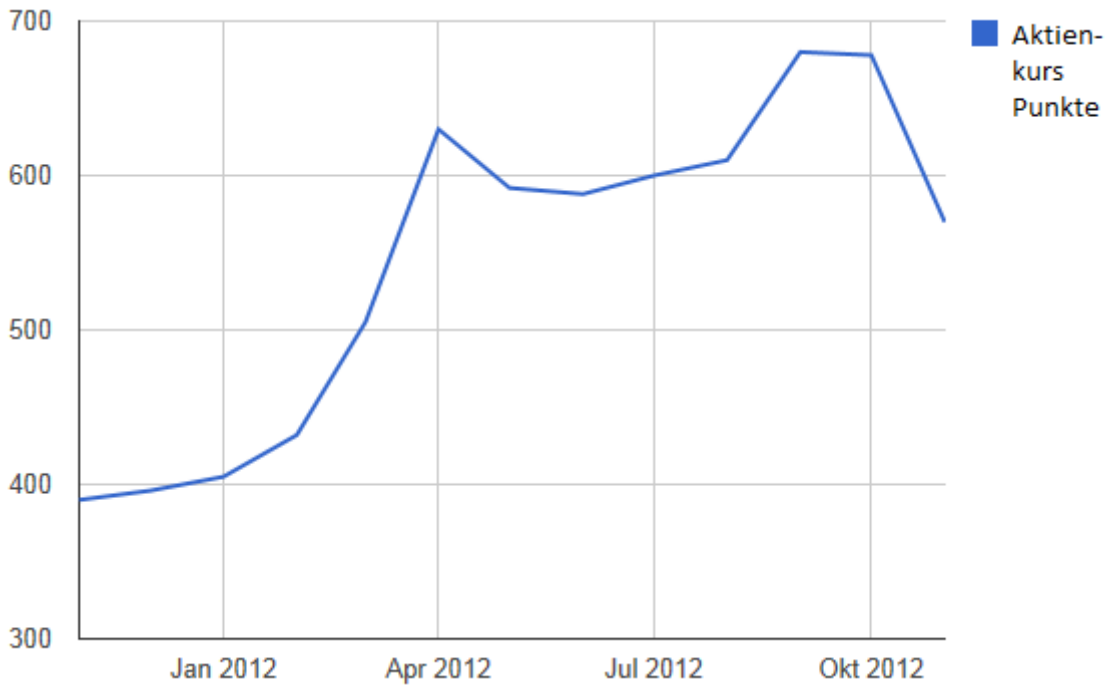

Abbildung 6.1: Apple Aktienkurs vom 1.11.2011 bis 1.10.2012

Zur Suche von Events eingegebene Ressourcen:
Apple Inc. - dbpedia.org/resource/Apple_Inc.

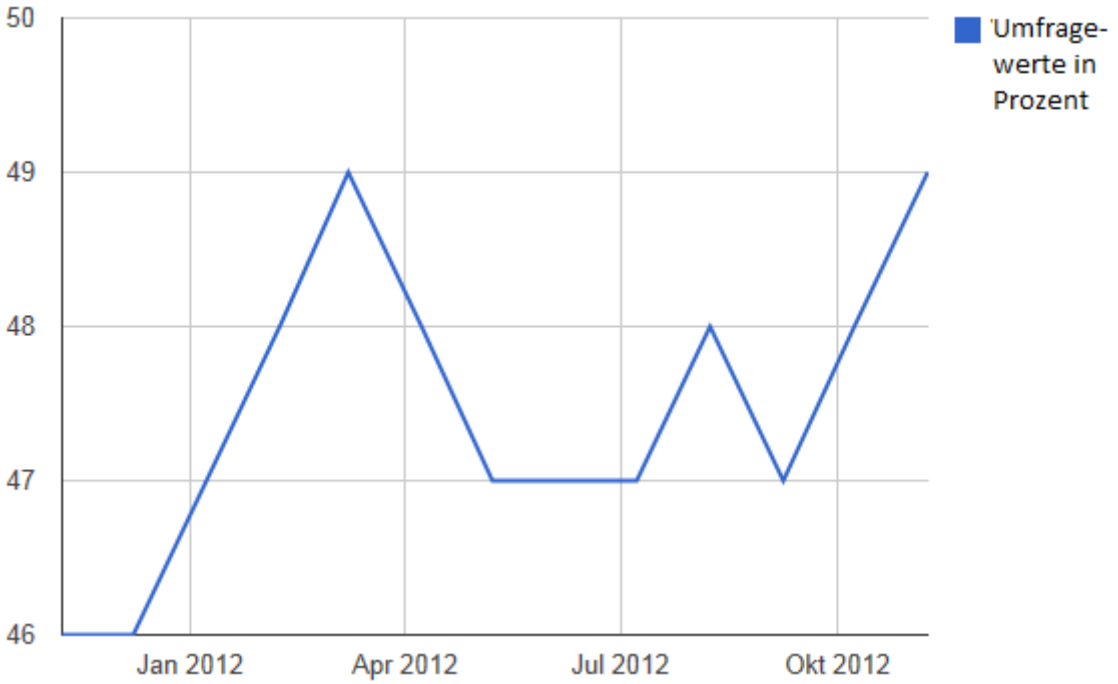

Abbildung 6.2: Obama Umfragewerte vom 8.11.2011 bis 8.10.2012

Zur Suche von Events eingegebene Ressourcen:
Barack Obama - http://dbpedia.org/resource/Barack_Obama

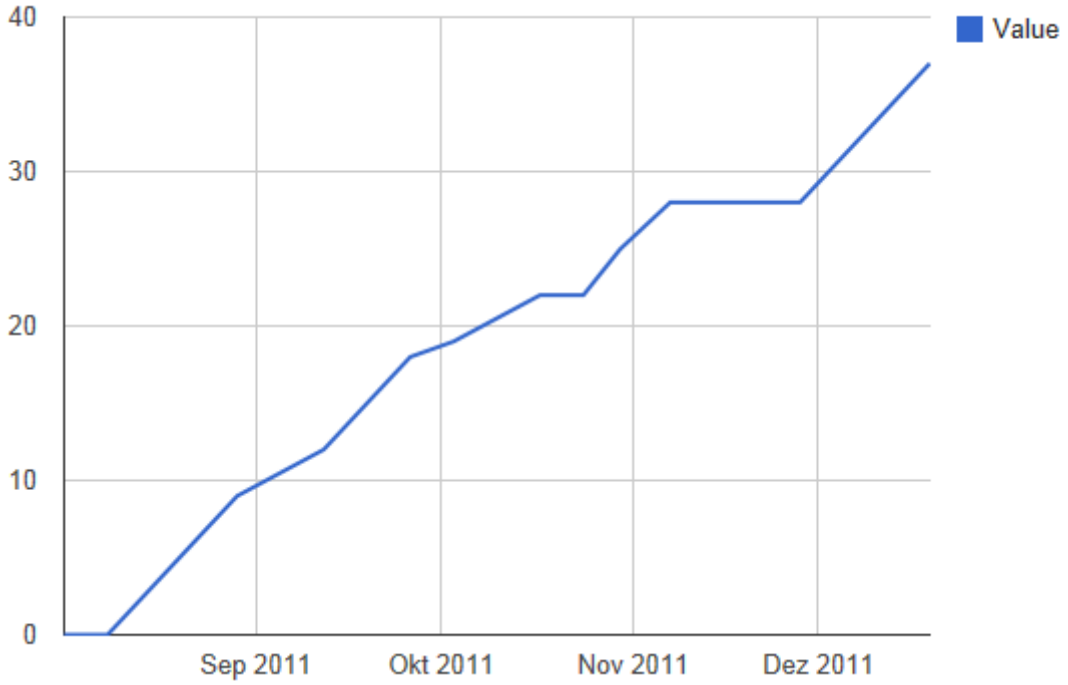

Abbildung 6.3: Punkteverlauf von Bayern München in der Hinrunde 2011

Zur Suche von Events eingegebene Ressourcen:
FC Bayern - http://dbpedia.org/resource/FC_Bayern_Munich

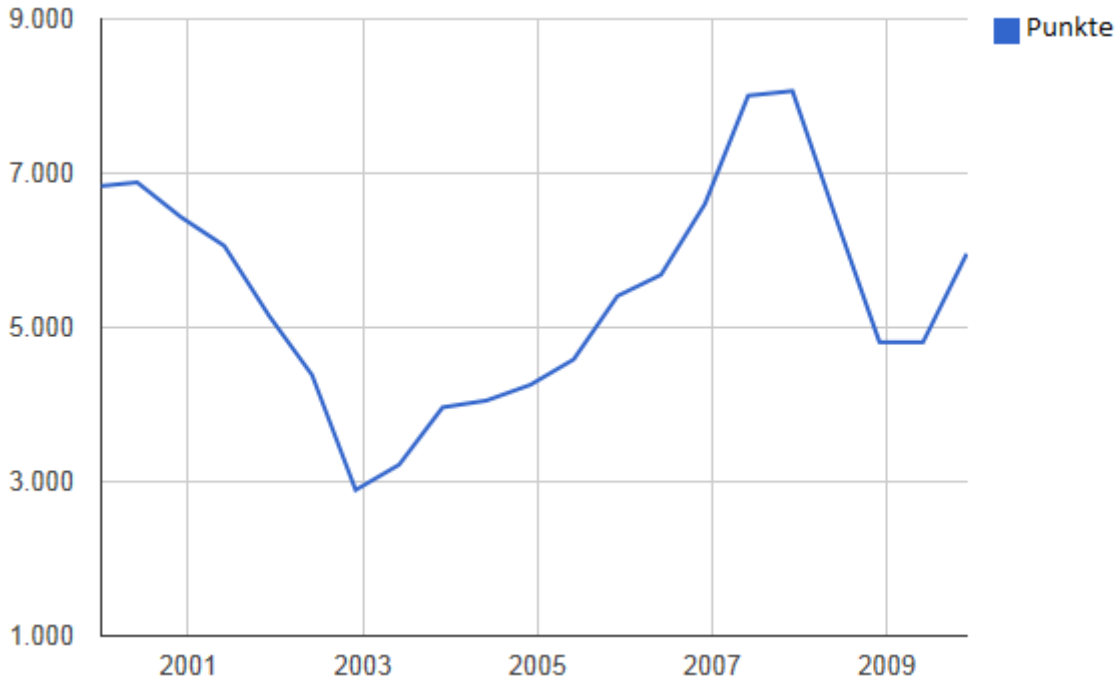

Abbildung 6.4: Punkteverlauf des DAX vom 3.1.2000 bis 1.12.2009

Zur Suche von Events eingegebene Ressourcen:
Finance - http://dbpedia.org/resource/Finance
DAX - http://dbpedia.org/resource/DAX
Stock Market - http://dbpedia.org/resource/Stock_market

Für jede Methode wird jeder Graph hinsichtlich Recall und von Hand auf Precision untersucht. Da es nicht ohne Weiteres möglich ist, zu ermitteln, wie viele Events es in einem bestimmten Zeitraum zu einem bestimmten Thema gibt, errechnet sich der Recall einfach aus der Anzahl gefundener Events in diesem Zeitraum. Findet das Programm zum Beispiel 20 Events im Zeitraum 1.10.2011-1.11.2011 so ist der Recall 20.
Die Precision wird nochmals aufgeteilt in *Passt zum Thema und erklärt* und *Passt zum Thema aber erklärt nicht*. Unter *Passt zum Thema und erklärt* fallen alle Events, die zum Thema passen und gleichzeitig den Verlauf des Graphen erklären. Unter *Passt zum Thema aber erklärt nicht* fallen alle Events, die zum Thema passen, den Verlauf des Graphen aber nicht erklären können oder sogar konträre Aussagen treffen. Wird allgemein von der Precision geredet, so sind damit Events gemeint, die zum Thema passen, egal ob sie den Verlauf erklären oder nicht.

Beispiel: Die Apple Aktie steigt signifikant. In diesem Zeitraum findet sich das Event ”Ein Drohnenangriff tötet vier Terroristen”. Dieses Event hat nichts mit dem Thema zu tun. Ein weiteres Event ist ”Apple vermeldet Rekordgewinne”. Dieses Event hat mit dem Thema zu tun und erklärt den Graphen, zählt also zu *Passt zum Thema und erklärt*. Ein weiteres Event ist ”Apples Zulieferfirma wegen schlechter Arbeitsbedingungen in der Kritik”. Dieses Event hat zwar mit dem Thema zu tun, erklärt den Verlauf des Graphen aber nicht beziehungsweise ist konträr zum Verlauf. Dieses Event zählt also zu *Passt zum Thema aber erklärt nicht*. Die letzten beiden Events zählen beide zur Precision.
Die Ergebnisse aus der Analyse werden in jedem Abschnitt zusammengefasst und allgemeine Schlüsse gezogen. Gibt es Ergebnisse zu einem Graphen, die sich gegenüber den anderen hervorheben, zum Beispiel durch besonders hohe Precision im Vergleich zu den anderen Graphen, so wird dies im Detail besprochen.

6.1 Simple Methode: Nur gegebene Ressourcen

6.1.1 Ohne Filter

Berücksichtigt man nur die gegebenen Ressourcen, ist der Recall generell sehr klein. Der Recall ist außerdem sehr stark vom Datenset und den gewählten Ressourcen abhängig. Ressourcen, die im Datenset selten vorkommen, senken den Recall enorm. Für den Apple-Graphen wurden zum Beispiel insgesamt nur sieben Events gefunden, auf zwölf Monate verteilt ergibt das eine durchschnittliche Eventanzahl von 0,64 pro Monat. Zum Obama-Umfragegraphen wurden dagegen 40 Events gefunden. Das ist verständlich, da eine Person wie Barack Obama, die als Präsident der USA sehr im öffentlichen Fokus steht, oft in den Nachrichten auftaucht. Noch mehr Events kann man mit Ressourcen wie `United_States` finden, da diese Ressource mit vielen Events verknüpft ist. Diese Aussage ist allerdings mit Vorsicht zu genießen. Werden Nachrichten wie etwa über die Präsidentschaftswahl in Amerika nicht mit `United_States` verbunden, sinkt auch hier der Recall. Ein Extremfall ist der FC-Bayern-Graph. Da das Datenset kaum Sportevents enthält, wurde kein einziges Event in Verbindung mit dem Thema FC Bayern gefunden. Generell ist der Recall also sehr niedrig, wobei es dennoch eine hohe Schwankung je nach gewählten Ressourcen gibt und man stärker von der Verlinkungsstruktur des Datensets abhängig ist.

Sehr gut ist die Precision bei diesem Ansatz. Das ist offensichtlich, denn jedes mit einer Ressource verlinkte Event hat direkt mit dieser Ressource zu tun. Wählt man also nur die Events aus, die mit den vorher ausgewählten Ressourcen in Verbindung stehen, ist eine hohe Genauigkeit zu erwarten. Nicht auszuschließen ist allerdings, dass gefundene Events mit dem Thema zu tun haben, den Verlauf des Graphen aber nicht erklären oder eine konträre Aussage treffen.

Das ist zum Beispiel beim Obama-Graphen der Fall. Eine gefunde Nachricht handelt von Nicolas Sarkozy, der bei einem Treffen mit Obama negativ über Israels Premierminister spricht. Diese Nachricht hat definitiv mehr mit Nikolas Sarkozy als Obama zu tun und keinen unmittelbaren Einfluss auf die Umfragewerte Obamas. Trotzdem wird dieses Event gefunden, da Sarkozys Aussage im Gespräch mit Obama fiel. Dies ist ein Beispiel dafür, dass auch bei dieser Methode, die nur die direkt eingegebenen Ressourcen zur Suche berücksichtigt, Events nicht zwangsläufig zum Thema passen müssen oder den Graphverlauf erklären. Da Obama eine Person der Öffentlichkeit wie kaum ein anderer ist, gibt es mehrere dieser Events, die zwar auch mit Barack Obama zu tun haben, aber nicht zum Thema ”Präsidentschaftswahl” passen.

Für Graphen wie den Apple-Aktienkurs ist dieses Problem nicht so groß. Da Apple eine Firma ist, haben alle Events, die zu dieser Ressource gefunden werden, direkt mit dem Stand der Firma zu tun und damit auch mit ihrem Aktienwert. Dazu gehören Nachrichten über einen Patentstreit mit Samsung ebenso wie Nachrichten über den Wert der Firma. In diesem Fall führt das dazu, das alle gefunden Events zum Thema passen. Nicht alle aber passen zum Verlauf der Graphen. So geht an einem Punkt der Verlauf des Graphen nach unten, es wird aber ein Event gefunden, das vom Wert der Firma handelt, der so hoch wie der keiner anderen auf der Welt sei. Das ist definitv eine positive Nachricht, der Trend des Graphen ist aber negativ, also passt das Event nicht zum Verlauf. Ähnlich verhält es sich mit dem DAX-Graph. Auch hier haben alle gefundenen Events mit dem Thema zu tun. Immerhin die Hälfte davon kann auch den Verlauf des Graphen erklären. Für Graphen aus dem Wirtschafts- und Finanzbereich scheint dieser Ansatz also recht gut zu funktionieren.

Die Precision ist außerdem komplett von den Eingaben des Nutzers abhängig. Wählt man bei einem Apple-Aktien-Graph zum Beispiel noch Ressourcen wie `Steve_Jobs` oder `Iphone` aus, so kann die Precision dadurch sinken. Gleiches gilt umgekehrt für den Recall, mehr vom Nutzer ausgewählte Ressourcen bedeuten einen höheren Recall. Ein weiterer Nachteil ist, dass der Nutzer nicht weiß, welche Ressourcen im Datenset bevorzugt zum Taggen von Events verwendet werden. Der Nutzer könnte also bei einem Graph über Umfragewerte von Barack Obama als Ressource lediglich `Barack_Obama` eingeben. Dadurch könnten ihm aber weitere relevante Events entgehen, die nur mit `Präsidentschaftswahl_Amerika` oder ähnlichem getaggt sind. Dies ist auch der Hauptkritikpunkt an diesem Ansatz: Er ernennt den Nutzer zum Experten, der nicht nur wissen muss, welche Ressourcen mit der Hauptressource in Verbindung stehen, sondern auch, mit welchen Ressourcen das Datenset seine Events bevorzugt taggt.

6.1.2 Mit SubjectBroader Filter

Der Filter ändert offensichtlich nichts an den Ergebnissen, da nur die Ressourcen verwendet werden, die vom Nutzer eingegeben wurden. Aus diesen Ressourcen werden überhaupt erst die Kategorien zum Filtern gebildet. Der Filter wird also kein Event ausschließen.

6.1.3 Mit UClassify Filter

Da der Filter hier nur auf der Basisi sehr weniger Events arbeitet, kann der Einfluss auf die Precision sehr hoch sein. Für den Apple-Graphen wird ein Event herausgefiltert. Hier geht es um eine nicht zugelassene Klage von Apple. Die News ist negativ, wird auch als solche eingestuft und daher herausgefiltert, da sie nicht zum Verlauf passt. Falsch eingestuft wird die Meldung, dass Apple die wertvollste Firma der Welt ist. Diese Meldung wird als negativ eingestuft und wird, da auch der Trend zu diesem Zeitpunkt negativ ist, nicht herausgefiltert. Falsche Filterungen geschehen aber nicht, deshalb steigt die Precision.

Auf den Obama-Graphen hat der Filter eine sehr große Auswirkung. Über die Hälfte aller Events werden herausgefiltert, übrig bleiben noch 17. Von den 23 herausgefilterten passen allerdings sieben Events zum Thema und können den Graphverlauf erklären. Das heißt, die Hälfte aller Events, die zutreffen, werden herausgefiltert. Am stärksten tritt dies in der Mitte der Graphen auf. Alle sechs Nachrichten des Intervalls werden herausgefiltert, obwohl drei davon den Verlauf erklären. Eine der herausgefilterten Nachrichten ist ”Obama unterstützt Same-Sex-Marriage”. Diese Nachricht ist kontrovers und polarisiert, passt also zum neutralen Graphenverlauf. Sie wird allerdings als sehr negativ eingestuft. Von den nicht zutreffenden werden insgesamt immerhin zwei Events herausgefiltert. Ansonsten werden vor allem Events herausgefiltert, die nicht mit dem Thema zu tun haben. Dadurch, das mehr zutreffende als nicht zutreffende Events herausgefiltert werden, steigt der Prozentsatz von *Passt zum Thema aber erklärt nicht*. Der Filter erreicht also genau das Gegenteil vom gewünschten Ergebnis.

Im Verhältnis sehr stark gefiltert wird beim DAX-Graphen. Nur noch zwei Events bleiben übrig. Das heißt, eines der Events, das zum Thema passt und den Graphen erklärt, wurde herausgefiltert. Bei der Nachricht handelt es sich um den Dollarkurs. Die restlichen drei herausgefilterten Events passten allesamt zum Thema, aber nicht zum Verlauf. In dieser Hinsicht hat der Filter hier sehr gut funktioniert.

	Apple	Obama	Bayern	DAX
Recall insgesamt	7	40	0	6
Recall pro Intervall	0,64	3,64	0	0,3
Passt zum Thema und erklärt	3 (0,43)	14 (0,35)	0	3 (0,5)
Passt zum Thema aber erklärt nicht	4 (0,57)	9 (0,225)	0	3 (0,5)

Tabelle 6.1: Nur gegebene Ressourcen - Ohne Filter

	Apple	Obama	Bayern	DAX
Recall insgesamt	6	17	0	2
Recall pro Intervall	0,54	1,54	0	0,1
Passt zum Thema und erklärt	3 (0,5)	7 (0,41)	0	2 (1)
Passt zum Thema aber erklärt nicht	3 (0,5)	7 (0,41)	0	0

Tabelle 6.2: Simple Methode: Nur gegebene Ressourcen - Mit UClassify Filter

6.2 InOut Methode

6.2.1 Ohne Filter

Berücksichtigt man zusätzlich zu den gegebenen Ressourcen alle Ressourcen, die mit ihnen in Verbindung stehen, erhöht sich der Recall stark. Im Beispiel des Obama-Umfragegraphen bedeutet dies über 1000 neue Ressourcen, die bei der Suche nach Events berücksichtigt werden. Das führt insgesamt zu durchschnittlich 21 gefundenen Events pro Zeitintervall, mehr als sechs Mal so viel gegenüber der *Simple Methode*. Ein Großteil der hinzugekommenen Events hat mit den USA zu tun, werden also gefunden, weil sie mit der Ressource `United_States` getaggt sind. Diese Ressource wurde mit in die Suche aufgenommen, da sie über einen Link mit `Barack_Obama` verbunden ist. Auch Nachrichten zu Vizepräsident Joe Biden werden nun gefunden, weitere interessante Ressourcen bei der Suche sind zum Beispiel `White_House`.

Im Falle des FC-Bayern-Graphen ändert sich dagegen nichts. Weiterhin werden keine Events gefunden. Das liegt daran, dass von der ursprünglichen Ressource `FC_Bayern_Munich` aus kaum allgemeinere Ressourcen gefunden werden, wie es zum Beispiel bei `Barack_Obama` mit `United_States` der Fall war. Die meisten Ressourcen, die gefunden werden, handeln von Spielern oder Managern des Clubs. Es gibt auch einen Link zur Ressource `Fußball_Bundesliga`, doch da das verwendete Datenset wie bereits erwähnt kaum Sportevents enthält, wird nichts gefunden.

Es kommt also immer auf die Ressource an, von der aus alle Verlinkungen gesucht werden. Eine Ressource mit sehr wenigen Verlinkungen auf ebenfalls weniger genutzte Ressourcen erhöht den Recall nur unmerklich. Sehr ausschlaggebend sind auch die Ressourcen, die hinzukommen, und wie oft diese in Events auftauchen.

Die Precision bei diesem Ansatz ist nicht sehr hoch. Durch die zusätzlichen Ressourcen werden viele Events gefunden, die mit dem Thema nichts zu tun haben. Beispiel: Dadurch, dass `United_States` als Ressource beim Obama-Umfragegraph hinzukommt, werden Events gefunden, die sich nur allgemein auf die USA beziehen, aber nichts mit der Präsidentschaftswahl zu tun haben - zum Beispiel eine Nachricht zum Nationalfeiertag. Andererseits werden durch diese Ressource auch Events gefunden, die mit dem Graph zu tun haben, etwa, dass sich die Lage auf den Arbeitsmarkt entspannt - eine Nachricht, die Obama zu seinem Vorteil nutzen konnte. Insgesamt bringt die Ressource `United_States` aber zu viele unnütze Events. Andere gefundene Ressourcen wie `White_House` dagegen haben prozentual öfter mit dem Thema zu tun.

Noch schlechter sieht es im Falle des Apple-Graphen aus. Auch die Ressource `Apple_Inc.` enthält über das Prädikat `locationCountry` einen Link auf `United_States`. Fast alle so gefundenen Events passen also überhaupt nicht zum Thema. News zum Holiday Online Shopping und der Wirtschaftslage passen, sonst geht es vor allem um Politik. Bis auf drei Ausnahmen dieser Kategorie kommen ausschließlich Events hinzu, die nicht zum Thema passen. Fast alle neu gefundenen Events sind aufgrund der Ressource `United_States` gefunden worden, lediglich Events wie ein Einbruch bei Steve Jobs, der mit der Ressource `Steve_Jobs` verbunden ist, bilden die Ausnahme. Das alles führt dazu, dass die Precision für den Apple-Graphen gegenüber dem Obama-Graphen schlechter ist. Das war bei der *Simple Methode* noch umgekehrt.

Sehr gut funktioniert die InOutMethode für den Dax-Graphen. Da der Graph sehr allgemein Aussagen über die Wirtschaft trifft, passen auch viele News, die sich entweder allgemein mit der Wirtschaftslage oder der Geschäftslage von Firmen im Speziellen beschäftigen. Weiterhin sind ein Großteil der InOutLinks immer noch eng mit dem Thema Wirtschaft verbunden, Links auf zu allgemeine Ressourcen wie `United_States` werden nicht gefunden. Die meisten gefundenen Ressourcen sind aus dem Bereich Stock Markets, Banken oder Firmen. Die neu gefundenen Ressourcen führen dazu, das in einem Zeitintervall 13 neue Events gefunden werden - vorher waren es null. Alle 13 Events haben mit dem Thema zu tun, zehn davon können den Graphverlauf sogar erklären. Das Intervall ist das des Börsencrashs 2008, gefundene Events handeln zum Beispiel von fallenden Kursen oder der Automobilindustriekrise. Gut zu sehen ist anhand des Graphen auch, dass das verwendete Datenset erst ab dem Jahr 2006 genügend Events enthält, um Erklärungen für Graphen zu liefern. Der Graph reicht vom Jahr 2000 bis 2010, bis zum Jahr 2006 werden aber nur drei Events gefunden, die restlichen 36 folgen danach.

Es scheint, je allgemeiner die Ressource, desto weniger nützliche Events werden durch sie gefunden im Verhältnis zur Anzahl neu gefundener Events. Es kommt aber auch auf den Graphen an, der betrachtet wird. Je allgemeiner der Graph, desto eher können sowohl allgemein gefasste Events, als auch speziellere Events den Verlauf erklären, so geschehen beim DAX-Graphen. Auffallend ist, dass *Passt zum Thema und erklärt* allgemein sehr viel höher im Vergleich zu *Passt zum Thema aber erklärt nicht* ist. Gefunde Events, die mit dem Thema zu tun haben, erklären den Graphen also meistens.

6.2.2 Mit SubjectBroader Filter

Für den Apple-Graphen werden durch den Filter zwei Drittel aller Events wieder herausgefiltert. Das erhöht die Precision stark, denn es werden keine Events herausgefiltert, die mit dem Thema zu tun haben. Dennoch bleiben viele themenfremde Events erhalten. Das Hauptproblem ist hier, dass über die Broader-Struktur eine Verbindung zwischen den Kategorien, die bei `United_States`, und den Kategorien, die bei `Apple_Inc.` gelistet sind, hergestellt werden kann. Das heißt, dass nach wie vor alle Events zu `United_States` aufgenommen werden. Die gemeinsame Kategorie ist `Article_Feedback_5_Additional_Articles`. Diese Kategorie wurde anscheinend automatisch aus Wikipedia extrahiert und besagt lediglich, dass alle Ressourcen dieser Kategorie zum Test von Wikipedias "Article Feedback Tool Version 5" selektiert wurden. Artikel, die dafür ausgewählt wurden, haben thematisch nichts miteinander zu tun, doch dbpedia erstellt automatisch eine gemeinsame Kategorie für die aus den Artikeln generierten Ressourcen. Interessanterweise werden dagegen Events, die mit der Ressource `California` getaggt sind, herausgefiltert, weil hier keine Verbindung hergestellt werden kann, und das obwohl Apple seinen Sitz in Kalifornien hat. Hieran sieht man gut, dass die Kategorien-Struktur von dbpedia nicht unbedingt gut ist, um gemeinsame Themengebiete festzustellen.

Ähnlich sieht es für den Obama-Graphen aus. Hier werden im Verhältnis allerdings nicht sehr viele Events herausgefiltert. Vorher wurden 233 gefunden, jetzt sind es noch 146. Alle herausgefilterten Events haben nichts mit dem Thema zu tun, die Precision erhöht sich im Verhältnis also stark. Für diesen Graphen funktioniert der Filter in dieser Hinsicht also gut. Allerdings wurden immer noch viel zu wenige Events herausgefiltert, es gibt nach wie vor zahlreiche themenfremde Events.

Der Fall, dass zu viele Events herausgefiltert werden, tritt beim Dax-Graphen ein. Insgesamt sieben Events, die zum Thema gehören und den Graphen erklären, wurden herausgefiltert. Dazu gehören Nachrichten über Turbulenzen am russischen Aktienmarkt und Nachrichten zur Krise in der Automobilindustrie. Würde man mit dem Filter noch einmal den `broader`-Link verfolgen und damit noch allgemeinere Kategorien miteinbeziehen, so würde man nach wie vor alle diese Events finden. Das ginge aber auf Kosten anderer Graphen, da dort dann auch allgemeinere Events miteinbezogen werden, was das Ergebnis im Falle des Apple-Graphen nur noch weiter verschlechtern würde. Dennoch funktioniert der Filter für den DAX-Graphen am besten, da auch sehr viele Events, die nichts mit dem Thema zu tun haben, herausgefiltert werden. Dadurch steigt die Precision.

Insgesamt zeigt sich, dass der *SubjectBroader Filter* nicht immer gut arbeitet. Im Falle des Apple-Graphen bleibt die sehr allgemeine Ressource `United_States` in der Suche drin und es werden daher kaum Events herausgefiltert. Im Falle des DAX-Graphen hingegen wird zu viel gefiltert, da sich zum Beispiel keine Verbindung zwischen dem russischen Aktienmarkt und den ursprünglichen Ressourcen herstellen lässt. Würde man im Falle des DAX-Graphen auch alle Überkategorien der bereits gefundenen Überkategorien mit einbeziehen, so würden keine relevanten Nachrichten herausgefiltert werden. Umgekehrt dürfte man beim Apple-Graphen gar keine Überkategorien mit einbeziehen, um bessere Werte zu erhalten. Den SubjectBroader Filter so einzustellen, dass er für alle Graphen gut funktioniert, ist also nur schwer möglich.

6.2.3 Mit UClassify Filter

Für den Apple-Graphen funktioniert der *UClassify Filter* nicht sehr gut. Es wird nur eines der Events herausgefiltert, das zum Thema passt, den Verlauf aber nicht erklärt. Die Nachricht über das Scheitern von Verhandlungen über Einsparungen im US-Haushalt etwa wird als sehr gut eingestuft. Gut ist dagegen die Einschätzung der Nachricht von Gesprächen von Apple mit China, um Probleme wie die Arbeitsbedingungen zu klären. Auch die anderen Events, die zum Thema passen und den Graphverlauf erklären, werden korrekt eingestuft. Dadurch werden keine dieser Events fälschlicherweise herausgefiltert. Dass die Precision insgesamt steigt, liegt schlussendlich hauptsächlich daran, dass viele Events herausgefiltert werden, die nicht zum Thema passen.

Hier tritt im ersten Intervall ein Fall auf, den man weniger dem Klassifizierer als dem Vorgehen des Filters anlasten muss. In diesem Intervall gibt es zwei negative und zwei positive Nachrichten. In Kombination können diese den neutralen Graphverlauf erklären. Der Klassifizierer schätzt eine der Nachrichten als sehr positiv, eine als sehr negativ ein. Das ist in Ordnung. Allerdings ist die Regel *Neutraler Graphverlauf = sehr positive/negative Events herausfiltern*. Hier führt also die aufgestellte Regel und nicht der Klassifizierer zu einem Fehler. Umgekehrt passiert es später, dass eine Nachricht nicht negativ genug eingeschätzt wurde und daher nicht herausgefiltert wird. Auch hier liegt der Klassifzierer richtig, nur die Regel führt zu einem Fehler. Insgesamt werden mehr Events korrekt gefiltert, wodurch die Ergebnisse sich leicht verbessern. Auch hier fallen als Nebeneffekt viele Events weg, die nicht zum Thema passen.

Im Falle des DAX-Graphen werden von 39 gefundenen Events 18 wieder herausgefiltert. Unter den herausgefilterten Events finden sich auch ein paar Events, die zum Thema passen und den Verlauf erklären können. Insgesamt steigt der Prozentsatz von *Passt zum Thema und erklärt* aber an. Der Großteil der herausgefilterten Events findet sich im Intervall, in dem ursprünglich 13 Events gefunden wurden. Hier werden sieben wieder herausgefiltert. Darunter befinden sich alle drei Events, die zum Thema passen, den Verlauf aber nicht erklären können, allerdings auch vier Events, die zum Thema passen und den Verlauf erklären. Hier ist auch eine interessante Klassifizierung zu beobachten. Die Nachricht "Trading is suspended for the third day in succession on Russia's two main stock exchanges, the MICEX and the dollar-denominated RTS, amidst fear of financial collapse. (...)" wird zu Recht als negativ eingestuft, die fast inhaltsgleiche Nachricht "Trading is suspended for the second day in succession on Russia's two main stock exchanges (the MICEX and the dollar-denominated RTS) after shares fall dramatically, (...)" jedoch fälschlicherweise als positiv.

6.2.4 Mit SubjectBroader und UClassify Filter

Bei dieser Methode wurde zunächst der *SubjectBroader Filter* angewandt, auf das Ergebnis anschließend der *UClassify Filter*.

Für den Apple-Graphen konnte der Recall gegenüber dem des *SubjectBroader*-Filters noch einmal leicht gesenkt werden. Insgesamt bleiben 36 Events übrig. Beide Filter in Kombination zahlen sich hier aus. Der SubjectBroader Filter nahm keine Events heraus, die mit dem Thema zu tun hatten. Auf diesem Ergebnis konnte der *UClassify Filter* wie auch schon vorher erfolgreich ein Event herausfiltern, das zum Thema passt, den Graphverlauf aber nicht erklären kann. Allerdings wurde auch ein Event herausgefiltert, das zum Thema passt und den Graphen erklärt. Außerdem wurden weitere Events herausgefiltert, die nicht zum Thema passen.

Das Gleiche gilt für den Obama-Graphen. Auch hier werden durch den *SubjectBroader Filter* nur Events herausgefiltert, die nicht zum Thema passen. Das Problem ist der *UClassify Filter*, der acht Events, die den Graphverlauf erklären, zu unrecht herausfiltert, gegenüber nur drei korrekt herausgefilterten Events. Weitere herausgefilterte Events passen nicht zum Thema. Der Recall ist jetzt nur noch 86, wodurch die prozentuale Precision insgesamt steigt.

Während sich die Filter bei dem Apple- und Obamagraphen gut ergänzen, tritt beim DAX-Graphen ein interessanter Fall ein. Der *Subject Broader Filter* und der *UClassify Filter* sortieren jeweils unterschiedliche Events heraus. Nur noch elf Events bleiben übrig. Das heißt, es wurden sehr viele Nachrichten herausgefiltert, die zum Thema passen und den Graphen erklären. Ungefiltert waren 20 Events in dieser Kategorie, jetzt sind es nur noch zehn. Dafür wurden aber auch alle Events herausgefiltert, die zum Thema passen, aber nicht den Verlauf erklären können. Das heißt, dass die nach Anwendung des *Subject Broader* Filters zwei überbleibenden Events, die zum Thema passen und den Verlauf nicht erklären können, vom *UClassify Filter* aussortiert werden. Insgesamt ergibt sich zwar eine Precision von 0,91, ein sehr guter Wert, allerdings auf Kosten zehn fälschlich herausgefilterter Nachrichten. Das Ergebnis ist hier gegenüber dem reinen *Subject Broader Filter* daher schlechter zu bewerten.

	Apple	Obama	Bayern	DAX
Recall insgesamt	161	233	0	39
Recall pro Intervall	14.64	21,18	0	1
Passt zum Thema und erklärt	8 (0,05)	23 (0,1)	0	20 (0,5)
Passt zum Thema aber erklärt nicht	8 (0,05)	9 (0,04)	0	7 (0,18)

Tabelle 6.3: InOut Methode - Ohne Filter

	Apple	Obama	Bayern	DAX
Recall insgesamt	50	146	0	18
Recall pro Intervall	4,5	13,2	0	0,9
Passt zum Thema und erklärt	8 (0,16)	24 (0,16)	0	14 (0,78)
Passt zum Thema aber erklärt nicht	8 (0,16)	9 (0,06)	0	2 (0,1)

Tabelle 6.4: InOut Methode - Mit SubjectBroader Filter

	Apple	Obama	Bayern	DAX
Recall insgesamt	111	134	0	21
Recall pro Intervall	10,1	13	0	1,05
Passt zum Thema und erklärt	7 (0,063)	16 (0,12)	0	13 (0,62)
Passt zum Thema aber erklärt nicht	7 (0,063)	6 (0,04)	0	3 (0,14)

Tabelle 6.5: InOut Methode - Mit UClassify Filter

	Apple	Obama	Bayern	DAX
Recall insgesamt	36	86	0	11
Recall pro Intervall	3,3	7,8	0	0,55
Passt zum Thema und erklärt	7 (0,19)	16 (0,19)	0	10 (0,91)
Passt zum Thema aber erklärt nicht	7 (0,19)	6 (0,07)	0	0

Tabelle 6.6: InOut Methode - Mit UClassify und Subject Broader Filter

6.3 Ressourcen aus gefundenen Events Methode

6.3.1 Ohne Filter

Auch bei dieser Methode erhöht sich der Recall stark. Der Recall ist sogar etwas höher im Vergleich zur *InOutMethode*. Das liegt vor allem daran, dass dieser Ansatz sehr auf die bevorzugten Taggings des Datensets eingeht. Bei Events gefundene Ressourcen werden mit einer sehr hohen Wahrscheinlichkeit auch zum Taggen von anderen Events verwendet. Das sieht man zum Beispiel an Hand des Obama-Graphen. Nur mit den gegebenen Ressourcen werden 40 Events gefunden, mit dieser Methode hingegen 985 Events. Das ist mehr als das Zwanzigfache. Eine noch höhere Verfielfachungsrate tritt beim Apple-Graphen auf. Hier werden 226 Events gefunden, ursprünglich waren es 7. Das ist mehr als das Dreißigfache. Umso erstaunlicher ist, dass bei der Suche nach weiteren Events für den Applegraphen nur 14 neue Ressourcen verwendet werden. Das zeigt, wie sehr der Recall erhöht werden kann, wenn man Ressourcen zur Suche verwendet, die im Datenset bevorzugt zum Taggen benutzt werden. Die absolute Anzahl an gefunden Events kann in einigen Fällen aber auch nur unmerklich steigen, nämlich dann, wenn zu den ursprünglichen Ressourcen kaum Events gefunden werden. Ein Extremfall ist hier der FC-Bayern-Graph. Es werden im ersten Durchlauf keine Events gefunden, folglich kann diese Methode auch keine weiteren Ressourcen in die Suche aufnehmen. Es werden also nach wie vor keine Events zu diesem Graphen gefunden.

Die Precision bei diesem Ansatz schwankt und ist insgesamt gering. Die Precision für den Applegraph verschlechtert sich im Vergleich zur *InOutMethode* kaum, da bei beiden Ansätzen gleich viele Events zum Thema passen und den Graphen erklären, insgesamt aber mehr Events gefunden werden (226 gegenüber161 mit der *InOutMethode*). Für den Obama Umfragegraph hingegen ist die Precision sehr gering, was vor allem dem sehr hohen Recall geschuldet ist. Viele der Ressourcen, die zur Suche hinzugenommen wurden, haben nichts mit dem Thema US-Walhkampf zu tun. So kommen alleine durch die Nachricht "The President of the United States Barack Obama and the Prime Minister of the United Kingdom David Cameron meet at the White House in Washington D.C. to discuss Afghanistan, Syria, the global economy and Iran." die Ressourcen `Iran`, `Syrien` und `Afghanistan` hinzu. Das führt im Falle von Syrien dazu, dass alle Events über die Kämpfe zwischen der Regierung und den Rebellen mit in den Graphen aufgenommen werden - das sind mehr als 50 Nachrichten alleine in den Monaten April bis Juni (inklusive). Kein einziges dieser Events steht mit den Umfragewerten in Verbindung.

Hier ist ein allgemeinenes Problem des Ansatzes zu erkennen: Sobald man auf sehr allgemeine Ressourcen stößt, schnellt der Recall in die Höhe. Die Ressourcen müssen aber nicht unbedingt nützlich sein. Mit dieser Methode stößt man auch leicht auf Ressourcen, die nur im speziellen Fall des gefundenen Events miteinander in Verbindung stehen, sonst aber wenig Gemeinsamkeiten haben. Die Kombination aus beiden - wie im Falle Syrien - ist für die Precision katastrophal.

Auch im Falle des DAX-Graphen schnellt der Recall in die Höhe, 490 Events werden gefunden. Nur 36 davon passen zum Thema und können den Graphverlauf erklären. Anders als beim Obama- oder Applegraphen sind die Events nicht fast identisch mit den gefundenen Events aus der *InOut-Methode*. Zum einen werden viele Events gefunden, die vorher nicht auftauchten, andererseits werden auch fünf Events nicht mehr gefunden. Ein interessanter Punkt wird vor allem bei dieser Methode mit diesem Graphen deutlich. Da das Zeitintervall mit sechs Monaten sehr hoch ist, können die Nachrichten teils sehr konträr zum Kursverlauf sein. So ist etwa kurz vor dem Absturz des Kurses durch die Weltwirtschaftskrise noch ein kleiner positiver Trend. Schon in diesem Intervall werden zahlreiche sehr negative Events gefunden, die vorausahnen lassen, dass der Kurs demnächst fallen wird. Für den aktuellen Graphverlauf sind die meisten dieser Nachrichten aber unpassend.

Insgesamt ist von allen Graphen die Precision für den DAX-Graphen am höchsten. Wie auch bei der *InOut-Methode* gilt, dass gefundene Events, die mit dem Thema zu tun haben, den Graphen meistens erklären.

6.3.2 Mit SubjectBroader Filter

Ein gutes Ergebnis wird für den Apple-Graphen erzielt. Keines der Events, die zum Thema passen, wird wieder herausgefiltert. Dadurch steigt die Precision insgesamt stark an. Dass der Recall mit 76 immer noch etwas zu hoch ist, liegt an der Ressource `United_States`. Wie bereits in der Evaluation der InOut-Methode beschrieben, ist hier das Problem die gemeinsame Kategorie `Article_Feedback_5_Additional_Articles`. Gäbe es diese Kategorie nicht, würde der Recall nochmal um die Hälfte sinken. Der Filter hätte in diesem Fall dann sehr gut funktioniert.

Für den Obama-Graphen ist das Ergebnis mittelmäßig. Es werden lediglich knapp die Hälfte aller Events herausgefiltert. Es bleiben immer noch 452 Events übrig, von denen eine große Mehrheit nichts mit dem Thema zu tun hat. Das

Problem ist hier, dass man über die `broader`-Struktur viel zu schnell zu sehr allgemeinen Kategorien gelangt. Alle anderen Politiker, seien sie nun direkt in den Wahlkampf involviert oder nicht, sind zum Beispiel über die Kategorie `Living People` mit Barack Obama verknüpft. Zum Teil, zum Beispiel für die Ressourcen `Turkey` oder `United_Kingdom`, tritt auch der `Article_Feedback_5_Additional_Articles`-Fall auf. Positiv ist hervorzuheben, dass keines der Events, das mit dem Thema zu tun hat, herausgefiltert wurde.

Für den DAX-Graphen funktioniert der SubjectBroader Filter sehr gut. Von den 490 Events wird ein Großteil zu Recht herausgefiltert, es bleiben 89 Events übrig. Unter diesen finden sich 43 Events, die zum Thema passen, 26 davon können den Graphverlauf auch erklären. Das heißt, dass 10 Events dieser Art wieder herausgefiltert wurden, ursprünglich waren es noch 36. Dadurch wurden im Verhältnis viel mehr Events herausgefiltert, die zum Thema passen und den Verlauf erklären, als Events, die zum Thema passen, aber nicht den Verlauf erklären können. Auch hier gilt wie schon bei der *InOut-Methode*: Hätte man noch eine `broader`-Stufe hinzugenommen, wären keine Events, die passen, herausgefiltert worden. Insgesamt ist das Ergebnis aber in Ordnung. Viele der Events gehören zum Thema Wirtschaft. Sie können also nicht direkt den Graphverlauf erklären, sind aber nicht völlig themenfremd. Das Ergebnis ist also etwas besser, als es die reinen Werte vermuten lassen.

6.3.3 Mit UClassify Filter

Für den Applegraphen werden verhältnismäßig wenige Events herausgefiltert, es bleiben noch 147 übrig. Da die Events, die zum Thema passen und den Graphen erklären, die gleichen sind, die auch bei der InOut-Methode gefunden wurden, ändert sich hier wenig gegenüber dieser Methode. Keines der Events, die zum Thema passen und den Graphen erklären, werden herausgefiltert. Da die Events, die zum Thema passen, aber nicht den Graphverlauf erklären, ebenfalls bis auf eine Ausnahme die gleichen sind, ändert sich auch hier nicht viel. Das zusätzliche Event handelt von einer Niederlage im Patentstreit gegen Samsung und wird zu Recht herausgefiltert, da dieses Event während eines Anstiegs des Graphen gefunden wurde. Der niedrigere Recall tritt hier wieder als Nebeneffekt auf, was die Precision insgesamt erhöht.

Im Falle des Obama-Graphen werden fast die Hälfte aller Events herausgefiltert, es bleiben 564 Events übrig. Der Großteil davon hat nicht mit dem Thema zu tun. Von den ursprünglich 30 passenden Events werden ganze 9 herausgefiltert. Das ist fast ein Drittel, ein unerwünschtes Ergebnis. Immerhin werden knapp die Hälfte aller Themen herausgefiltert, die zum Thema passen, aber nicht den Graphverlauf erklären. Hier bleiben noch 8 Events übrig. Es scheint, als könne der *UClassify Filter* Politiknachrichten weniger gut einschätzen. Zusätzlich gibt es Nachrichten, die negativ sind, gerade deshalb aber Obama begünstigen. So wird zum Beispiel eine negative Nachricht über den Campaign-Manager seines Konkurrenten Mitt Romney korrekterweise als negativ eingestuft. Für Obama ist diese Nachricht allerdings positiv und kann so den Graphverlauf erklären. Das kann der *UClassify Filter* natürlich nicht wissen, weshalb dieses Event herausgefiltert wird. Hier tritt also ein grundlegendes Problem auf: Die Klassifizierer können nur den Text an sich einschätzen, und nicht für wen oder was dieser Text positiv oder negativ ist.

Ein sehr schlechtes Ergebnis ist für den DAX-Graphen zu beobachten. Lediglich drei der 17 Events, die zum Thema passen, den Graphverlauf aber nicht erklären können, werden herausgefiltert. Demgegenüber werden fälschlicherweise zwölf der 36 Events herausgefiltert, die zum Thema passen und den Verlauf erklären können. Das führt dazu, dass sich das Verhältnis von passenden und unpassenden Nachrichten verschlechtert statt wie gewünscht verbessert. Das ist umso überraschender, da die Klassifizierung von Wirtschaftsnachrichten sonst meistens sehr gut funktioniert hat. Einige Events werden negativ beziehungsweise positiv eingestuft, aber nicht negativ beziehungsweise positiv genug, um vom Filter herausgenommen zu werden. Hier liegen die Ursachen also auch im Vorgehen des Filters und nicht nur der Klassifizierung.

6.3.4 Mit Subject Broader und UClassify Filter

Bei dieser Methode wurde zunächst der *SubjectBroader Filter* angewandt, auf das Ergebnis wurde der *UClassify Filter* angewandt.

Für den Apple-Graphen funktioniert die Kombination beider Filter gut, weil der *Subject Broader Filter* keines der Events, die zum Thema passen, herausfiltert. Die Anzahl an unnützen Events wird also gesenkt. Auf diesem Ergebnis kann der *UClassify Filter* arbeiten und filtert dabei dieselben Events heraus, die zum Thema passen, wie schon vorher. Es wird also wieder fälschlicherweise ein Event herausgefiltert, das zum Thema passt und den Verlauf erklärt. Demgegenüber wird ein anderes Event korrekt herausgefiltert. Da nebenbei auch viele unnütze Events herausgefiltert werden, ist die Kombination beider Filter ein Gewinn und die Precision verbessert sich.

Das Gleiche gilt für den Obama-Graphen. Auch hier filtert der *Subject Broader Filter* keines der Events heraus, die zum Thema passen. Der *UClassify Filter* sortiert danach wieder fälschlicherweise 9 der 30 passenden Events heraus, die den Verlauf erklären. Korrekterweise werden 7 der 15 Events, die zum Thema passen, den Verlauf aber nicht erklären können, herausgefiltert. Da der *UClassify Filter* als Nebeneffekt weitere unnütze Events aussortiert, steigt die Precision insgesamt an, die Kombination beider Filter ist also auch hier ein Gewinn.

Für den DAX-Graphen tritt ein ähnlicher Fall auf, wie wir ihn schon bei *InOut Methode - Mit Subject Broader und UClassify Filter* beobachten konnten. Wieder sortieren die Filter fälschlicherweise unterschiedliche Events heraus, die zum Thema passen und den Verlauf erklären. Von ursprünglich 36 passenden bleiben mit dem *Subject Broader Filter* 26 übrig, mit dem *UClassify Filter* 24. Die Kombination beider Filter senkt zwar den Recall von ursprünglich 480 auf 38, was gut ist, filtert dabei aber auch viel mehr Events heraus, die zum Thema passen. Hier bleiben lediglich 14 Events übrig, die zum Thema passen und den Verlauf erklären. Der *Subject Broader Filter* sortiert also 12 passende Events aus, die der *UClassify Filter* nicht aussortiert hätte. Durch den niedrigen Recall steigt zwar die Precision, doch aufgrund vieler fälschlicherweise aussortierter Events ist die Kombination beider Filter nicht gut.

	Apple	Obama	Bayern	DAX
Recall insgesamt	226	985	0	490
Recall pro Intervall	20,55	89,55	0	24,5
Passt zum Thema und erklärt	8 (0,036)	30 (0,031)	0	36 (0,073)
Passt zum Thema aber erklärt nicht	8 (0,036)	15 (0,015)	0	21 (0,043)

Tabelle 6.7: Ressourcen aus gefundenen Events Methode - Ohne Filter

	Apple	Obama	Bayern	DAX
Recall	76	452	0	89
Recal pro Intervalll	6,91	41,1	0	4,45
Passt zum Thema und erklärt	8 (0,11)	30 (0,066)	0	26 (0,29)
Passt zum Thema aber erklärt nicht	8 (0,11)	15 (0,033)	0	17 (0,19)

Tabelle 6.8: Ressourcen aus gefundenen Events Methode - Mit SubjectBroader Filter

	Apple	Obama	Bayern	DAX
Recall	147	564	0	290
Recal pro Intervalll	13,36	51,27	0	14,5
Passt zum Thema und erklärt	7 (0,048)	21 (0,037)	0	24 (0,08)
Passt zum Thema aber erklärt nicht	7 (0,048)	8 (0,014)	0	14 (0,05)

Tabelle 6.9: Ressourcen aus gefundenen Events Methode - Mit UClassify Filter

	Apple	Obama	Bayern	DAX
Recall	48	267	0	38
Recal pro Intervalll	4,36	24,3	0	1,9
Passt zum Thema und erklärt	7 (0,15)	21 (0,079)	0	14 (0,37)
Passt zum Thema aber erklärt nicht	7 (0,15)	8 (0,03)	0	6 (0,16)

Tabelle 6.10: Ressourcen aus gefundenen Events Methode - Mit SubjectBroader und UClassify Filter

6.4 InOut Methode - Ordered by Relevance

6.4.1 Ohne Filter

Da bei diesem Ansatz alle Ressourcen wegfallen, die keinen gemeinsamen Themenbereich mit der ursprünglich betrachteten Ressource haben, schränkt sich die Anzahl der Ressourcen insgesamt sehr ein und dadurch auch der Recall. Für den Apple-Graphen wurden zur Suche nach Events nur noch 30 Ressourcen verwendet. Bei der *InOut Methode* waren es noch 603. Viele der Ressourcen sind auch spezieller als die ursprüngliche Ressource `Apple_Inc`. So kamen Ressourcen wie `Apple_Campus` oder `IPhone_4S` hinzu. Der Recall ist daher nicht viel höher im Vergleich zur Methode, nur die ursprünglichen Ressourcen zur Suche zu verwenden. Im Falle des Apple-Graphen wurden nur drei neue Events gefunden, im Falle des Obama-Graphen zehn neue, im Falle des DAX-Graphen ist es sogar nur ein neues Event.

Die neuen Events, die durch die weiteren Ressourcen hinzukommen, sind leider meistens nicht hilfreich. Das bedeutet, dass sich die Precision leicht verschlechtert. Erneut ist *Passt zum Thema und erklärt* sehr viel höher als *Passt zum Thema aber erklärt nicht*. Im Falle des Apple-Graphen werden zum Beispiel zwei Nachrichten über Microsoft aufgenommen, da auch die Ressource `Microsoft` bei der Suche nach Events verwendet wurde. In einer weiteren Nachricht geht es um einen Einbruch bei Steve Jobs, die wegen der Ressource `Steve_Jobs` gefunden wurde. Das heißt, dass keine der drei neuen Nachrichten gegenüber *Simple Methode* einen Vorteil bringen und im Gegenteil die Precision verschlechtern, da keine der News für den Graphen relevant sind. Besser sieht es schon für den Obama-Graphen aus. Einige Nachrichten passen zum Thema und können den Graphen sogar erklären. So werden zum Beispiel Nachrichten aufgeführt, die zwar nicht direkt mit Barack Obama zu tun haben, die ihm die Republikaner aber anlasten konnten und so Auswirkungen auf den Wahlkampf hatten. Dadurch steigt die Precision gegenüber der simplen Methode sogar leicht. Das neu gefundene Event für den DAX-Graphen passt ebenfalls zum Graphen und kann den Verlauf sogar erklären, also steigt auch hier die Precision.

Der Grund, warum kaum neue Events hinzukommen, ist im Falle des Apple-, DAX- und Obama-Graphen gut zu erklären: Alle Ressourcen, die hinzukommen, sind speziellere Ressourcen als die ursprünglich betrachtete. Die ursprüngliche Ressource `Apple_Inc.` ist zum Beispiel noch sehr allgemein und so bei einigen Events getaggt, die durch diese Methode hinzukommende Ressource `Iphone` hingegen ist schon spezieller und vereinigt wenige Events auf sich. Zusätzlich ist davon auszugehen, dass alle Events, die mit `Iphone` getaggt sind, auch mit `Apple_Inc.` getaggt sind. Die Methode ist in diesem Fall also kein Gewinn und nur dann hilfreich, wenn der User speziellere Ressourcen wie Iphone eingibt, worauf die Methode auch auf allgemeinere Ressourcen wie `Apple_Inc.` kommen wird. Ähnlich verhält es sich mit dem DAX- und dem Obama-Graphen. Auch hier kommen fast nur speziellere Ressourcen hinzu. Im Falle des Bayern-Graphen kommen Ressourcen anderer Bundesligaclubs hinzu, was theoretisch gut passen würde, doch werden hier keine Events im Datenset gefunden.

6.4.2 Mit SubjectBroader Filter

Durch den *SubjectBroader Filter* ändert sich nichts, kein Event wird herausgefiltert. Bei der *Ordered by Relevance* Methode werden nur Ressourcen in die Suche nach Events mit aufgenommen, die mindestens einen Themenbereich mit der ursprünglichen Ressource gemeinsam haben. Diese Kategorien sind natürlich auch in der Liste der Kategorien, die bei diesem Filter erstellt wird, enthalten, da diese ebenfalls von den Themenbereichen der ursprünglichen Ressource ausgehend erstellt wird.

6.4.3 Mit UClassify Filter

Durch den *UClassify Filter* fallen einige Events wieder weg, der Recall sinkt. Für den Obama-Graphen werden auf diese Weise nur noch 24 Events gefunden, das heißt über die Hälfte aller Events, also 26 Stück, werden herausgefiltert. Das gewünschte Ergebnis, den Anteil an Events, die zum Thema passen, den Graphen aber nicht erklären, zu senken, wird erreicht. Nur noch 0,17 aller Events gehören nun dieser Kategorie an. Das geht allerdings auch auf Kosten der Events, die zum Thema passen und den Graphen erklären. Hier sinkt der Anteil ebenfalls leicht, immerhin nur auf 0,38. Interessant zu beobachten ist, wie die Nachrichten eingestuft werden. So wird die Nominierung von Obama zum Präsidentschaftskandidaten als sehr negative Nachricht eingestuft. Aus diesem Grund taucht diese Nachricht auch weiterhin auf, da sie laut Filter mit dem negativen Trend konform ist. Tatsächlich ist diese Nachricht aber eine positive und kann den Graphen daher nicht erklären. An dieser Stelle schätzt der *UClassify Filter* also falsch. In einem Fall stuft der Classifier eine Nachricht, in der Obama neue Sanktionen gegen den Iran verhängt, als sehr negativ ein. Geht man nur von der Nachricht

an sich aus, stimmt das auch, doch auf den Wahlkampf wirkt sich diese Entscheidung positiv für Obama aus. Die Nachricht hätte daher auch mit dem positiven Trend des Graphen übereingestimmt, wird aber herausgefiltert, da sie als sehr negativ eingestuft wurde. Interessanterweise wird eine ähnliche Nachricht - wieder geht es um Sanktionen gegen den Iran - als sehr positiv eingestuft und bleibt daher, da sie auch den postiven Trend erklärt, erhalten. Viele Nachrichten, die überhaupt nicht mit dem Thema zu tun haben, werden durch den Filter ebenfalls herausgeworfen.

Für den Apple-Graphen funktioniert der Filter besser, hier werden nur zwei Events wieder herausgefiltert, insgesamt werden also acht Events gefunden. Eines der Events hat nichts mit dem Thema zu tun und wird daher mehr zufällig herausgefiltert. Die andere Nachricht geht um einen Patentstreit zu Ungunsten Apples. Diese Nachricht wird als negativ eingstuft, und da sie damit nicht dem Trend entspricht, herausgefiltert. Der Filter arbeitet insgesamt sehr gut, nur eine Nachricht wird falsch eingeschätzt. Hier geht es darum, dass Apple so viel wert ist wie keine andere Firma der Welt, diese Nachricht wird jedoch negativ eingestuft. Eigentlich hätte diese Nachricht herausgefiltert werden müssen, da der Trend negativ ist, doch da die Nachricht falsch eingestuft wurde, bleibt sie erhalten.

Hervorragend funktioniert der Filter für den DAX-Graphen. Hier werden alle drei Events, die zum Thema passen, den Graphen aber nicht erklären, herausgefiltert. Das geht auf Kosten eines weiteren Events, das auch zum Thema passt und den Graphen erklärt. Alle drei verbleibenden Events, die nicht gefiltert werden, passen zum Thema und erklären den Verlauf. Für Wirtschaftsnachrichten scheint der *UClassify Filter* also besonders gut zu funktionieren, da die Nachrichten oft einem Schema folgen und bestimmte Ausdrücke immer im selben Zusammenhang auftauchen.

	Apple	Obama	Bayern	DAX
Recall insgesamt	10	50	0	7
Recall pro Intervall	0,83	4,55	0	0,35
Passt zum Thema und erklärt	3 (0,3)	20 (0,4)	0	4 (0,57)
Passt zum Thema aber erklärt nicht	4 (0,4)	13 (0,26)	0	3 (0,43)

Tabelle 6.11: InOut Methode - Ordered by Relevance - Ohne Filter

	Apple	Obama	Bayern	DAX
Recall insgesamt	8	24	0	3
Recall pro Intervall	0,72	2,18	0	0,15
Passt zum Thema und erklärt	3 (0,38)	9 (0,375)	0	3 (1)
Passt zum Thema aber erklärt nicht	3 (0,38)	4 (0,17)	0	0

Tabelle 6.12: InOut Methode - Ordered by Relevance - Mit UClassify Filter

6.5 Über zwei Links verwandte Ressourcen - Order by Relevance

6.5.1 Ohne Filter

Wie auch bei *InOutMethode - Ordered by Relevance* fallen bei diesem Ansatz alle Ressorucen weg, die keinen gemeinsamen Themenbereich mit der ursprünglich betrachteten Ressource haben. Da allerdings Ressourcen, die über zwei Kanten mit der ursprünglichen Ressource verbunden sind, betrachtet werden, ist die Anzahl ungleich höher. Für die Ressource `Barack_Obama` zum Beispiel gibt es über 600 Ressourcen, die über zwei Kanten verbunden sind und gemeinsame Themenbereiche aufweisen. Unter den ersten 100 finden sich aber fast ausschließlich andere Politiker, eine allgemeinere Ressource wie `United_States` folgt erst später. So kommt es, dass der Recall trotz der vielen Ressourcen nicht so stark steigt, wie vielleicht zu erwarten ist. Dennoch ist der Recall im Vergleich zur *InOut Methode - Ordered by Relevance* noch einmal höher. Für den Obama-Graphen werden insgesamt 76 Events gefunden.

Die neu hinzugekommenen Events tragen leider nicht viel zur Erklärung des Graphen bei. Dadurch sinkt die Precision. Die gefundenen Events sind trotzdem noch eng mit dem Wahlkampf verbunden. Das liegt daran, dass über diese Methode vor allem politische Personen als neue Ressourcen aufgenommen werden. Dadurch werden auch Nachrichten über die Kandidaten der Republikaner gefunden. Diese haben zwar mit dem Thema US-Wahlkampf zu tun, können den Graphverlauf aber nicht direkt erklären.

Für den Apple-Graphen werden viele Nachrichten gefunden, die man im erweiterten Sinne dem Themenbereich von Apple zuschreiben kann. Das ist nicht verwunderlich, denn so ist diese Methode konzipiert. Leider werden dadurch kaum Events gefunden, die direkt mit Apple zu tun haben und den Graphen erklären können. Viele der Nachrichten handeln zum Beispiel von Facebook, da diese Ressource mit in die Suche aufgenommen wurde. Andere Nachrichten handeln von Google, Yahoo oder IBM. Hier ist es schwer zu sagen, welche der Nachrichten zumindest indirekt Einfluss auf den Apple-Aktienkurs haben. Sie werden hier als nicht zum Thema passend markiert, die Qualität der Nachrichten zu diesem Graphen sind aber unter Umständen sehr viel besser, als es die Werte vermuten lassen. Zumindest sind die Nachrichten ”nah dran”. Lediglich ein neues Event wird gefunden, das zum Thema passt und den Graphverlauf erklärt. Die Nachricht geht um die Eurokrise, die aufgrund der nachlassenden Kaufkraft natürlich auch Apple betrifft. Die Nachricht wird während eines Kursabfalls gefunden, kann den Verlauf also erklären.

Kaum Veränderung gibt es beim DAX-Graphen. Hier ist das Ergebnis das gleiche wie schon bei der *InOut Methode - Ordered by Relevance*. Dieselben zusätzlichen Ressourcen werden gefunden, was zu einem weiteren Event führt, das zum Thema passt und den Graphen erklärt. Hier steigt die Precision also leicht.

Der Grund, warum kaum neue Events hinzukommen, ist ähnlich zu erklären wie auch bei *InOut Methode - Ordered by Relevance*. Viele der gewonnenen Ressourcen sind sehr speziell und sind somit kaum als Tags bei Events zu finden. Wie schon oben beschrieben werden im Falle der `Ressource Barack_Obama` in erster Linie andere Politiker gefunden, viele davon weniger nachrichtenpräsent. Sobald aber allgemeine Ressourcen wie `United_States` hinzukommen, schnellt der Recall in die Höhe.

6.5.2 Mit SubjectBroader Filter

Durch den *SubjectBroader Filter* ändert sich nichts, kein Event wird herausgefiltert. Bei der *Ordered by Relevance* Methode werden nur Ressourcen in die Suche nach Events mit aufgenommen, die mindestens einen Themenbereich mit der ursprünglichen Ressource gemeinsam haben. Diese Kategorien sind natürlich auch in der Liste der Kategorien, die bei diesem Filter erstellt wird, enthalten, da diese ebenfalls von den Themenbereichen der ursprünglichen Ressource ausgehend erstellt wird.

6.5.3 Mit UCLassify Filter

Durch den *UClassify Filter* ändert sich für den Apple-Graphen verhältnismäßig wenig. Lediglich 13 Events werden wieder herausgefiltert. Eines der herausgefilterten Events hat mit dem Thema zu tun, kann den Graphen aber nicht erklären. Dass zwölf nicht zum Thema passende Events herausgefiltert werden und dadurch die Precision insgesamt ansteigt, ist mehr ein Nebeneffekt, da die Wahrscheinlichkeit, ein zutreffendes Event herauszufiltern, sehr gering ist. Insgesamt steigt dadurch die Precision an, wobei die Prozentzahl an Events, die zum Thema passen, den Graphen aber nicht erklären, gleich bleibt. Die Einschätzung der Nachrichten funktioniert meistens gut. Die vorher neu gefundene Nachricht über die Eurokrise zum Beispiel wird negativ eingestuft, weshalb sie nicht herausgefiltert wird, da sie den Kursverlauf erklären kann.

Ein Fall, in dem die Einschätzung nicht gut funktioniert, tritt bei einer Nachricht über Facebook auf. Facebook stelle automatisch ein, dass Nutzer eine at-facebook-Mail verwenden, diese Nachricht wurde fälschlicherweise als positiv eingestuft.

Für den Obama-Graphen sind die Ergebnisse ähnlich zu denen von *InOut Methode - Ordered by Relevance - Mit UClassify Filter*. Wie auch in jener Methode fallen über die Hälfte aller Events wieder heraus. Viele davon passen nicht zum Thema, wodurch die Precision steigt. Allerdings steigt vor allem die Prozentzahl von Events an, die zum Thema passen, den Verlauf aber nicht erklären können. So wird die Nominierung von Obama zum Präsidentschaftskandidaten fälschlicherweise als sehr negative Nachricht eingestuft. Aus diesem Grund taucht diese Nachricht auch weiterhin auf, da sie laut Filter mit dem negativen Trend konform ist. Für den Obama-Graphen führt der Filter also zum Gegenteil des gewünschten Ergebnisses.

Für den DAX-Graphen sind die Ergebnisse die gleichen wie schon in *InOut Methode - Ordered by Relevance - Mit UClassify Filter*. Das ist logisch, da auch ohne Filter genau die gleichen Events wie schon durch *InOut Methode - Ordered by Relevance* gefunden wurden. Wieder werden alle nicht zutreffenden Events herausgefiltert und die restlichen drei Events passen alle zum Thema und können den Graphverlauf erklären.

	Apple	Obama	Bayern	DAX
Recall	51	76	0	7
Recall pro Intervall	4,63	6,91	0	0,35
Passt zum Thema und erklärt	4 (0,08)	19 (0,25)	0	4 (0,57)
Passt zum Thema aber erklärt nicht	4 (0,08)	13 (0,17)	0	3 (0,43)

Tabelle 6.13: Über zwei Links verwandte Ressourcen - Order by Relevance - Ohne Filter

	Apple	Obama	Bayern	DAX
Recall	38	36	0	3
Recall pro Intervall	3,54	3,27	0	0,15
Passt zum Thema und erklärt	4 (0,11)	13 (0,36)	0	3 (1)
Passt zum Thema aber erklärt nicht	3 (0,08)	12 (0,32)	0	0

Tabelle 6.14: Über zwei Links verwandte Ressourcen - Order by Relevance - Mit UCLassify Filter

7 Fazit

Für den Apple-Aktienkurs funktionierte keiner der Ansätze besonders gut. Das beste Ergebnis wurde mit der *Simple Methode* und dem *UClassify Filter* erzielt. Der Filter konnte ein negatives Event korrekt herausfiltern. Alle anderen Methoden, die weitere Ressourcen zur Suche nach Events einbezogen, führten zu einer sehr schlechten Precision. Ein Problem ist hier, dass zum Beispiel allgemeine Events zur Wirtschafts- und Joblage in den USA an die Ressource `United_States` und nicht anderes Spezielleres gebunden waren. Dadurch wurden diese Events auf Kosten vieler unnützer Nachrichten gefunden. Die meisten Events wurden bei der *InOut-Methode* und der *Ressourcen auf gefundenen Events*-Methode gefunden. Der *Subject Broader Filter* und der *UClassify Filter* verbesserten immer die Precision. Für die meisten Methoden half der *Subject Broader Filter* sogar eher als der *UClassify Filter*, da hier mehr unnütze Events herausgefiltert wurden und so die Precision stärker anstieg. Bei der *Ressourcen aus Events*-Methode konnte die Eventzahl so von 226 auf 76 gebracht werden, der *UClassify* filterte nur auf 147 Events herunter. Die Kombination beider Filter erzielte ein noch besseres Ergebnis. Generell scheint das Problem für den Apple-Graphen, dass es sehr wenige Events gibt, die zum Thema passen. Das Datenset scheint nicht umfangreich genug, um solche speziellen Aktienkurse erklären zu können.

Für den Obama-Umfragegraphen funktionierten die beiden *Order by Relevance*-Methoden am besten. Fast alle gefundenen Events passten zum Themengebiet. Viele Nachrichten hatten mit den Republikanern und der Wahl ihres Kandidaten zu tun. Sie konnten den Graphverlauf daher nicht direkt erklären, die Ergebnisse sind aber besser als die reinen Zahlen vermuten lassen. Die Events sind sozusagen ”nah dran”. Das beste Ergebnis wurde ohne Filter erzielt. Der *UClassify Filter* konnte für den Obamagraphen nicht überzeugen. Es wurden zu viele Events herausgefiltert, die zum Thema passen und den Verlauf erklären. Gleichzeitig wurden zu wenige Events herausgefiltert, die den Verlauf nicht erklären können. Die Nachricht über die Nominierung Obamas als Kandidat etwa wurde fälschlicherweise als negativ eingeschätzt. Auch in anderen Fällen lag der Klassifizierer daneben. Der Filter funktionierte aber auch deswegen nicht so gut, weil die Regeln, nach denen gefiltert wurde, nicht funktionierten. So wurden einige Events zwar negativ, aber nicht negativ genug eingeschätzt, um herausgefiltert zu werden. Ein drittes Problem ist, dass Nachrichten wie etwa Sanktionen gegen den Iran korrekterweise als negativ eingestuft wurden, diese im Bezug auf Obama aber positiv sind. Die Sicht des Betrachters wurd also nicht mit einbezogen. Die Kombination beider Filter lieferte immerhin bessere Werte als die Filter im Einzelnen. Die meisten Events wurden mit der *Ressourcen aus Events*-Methode gefunden. Unter den fast 1000 Nachrichten war aber eine große Mehrheit unnütz, da Ressourcen wie `Syrien` oder `Afghanistan` hinzukamen.

Für den Bayerngraphen wurden mit keiner Methode Events gefunden. Bei einer manuallen Suche nach dem Stichwort ”Soccer” wurden 170 Treffer gefunden. Da das Wort durch Links in den Nachrichten mehrmals vorkommt, kann man von 80 Nachrichten, die direkt mit Fußball zu tun haben, ausgehen. Für das Stichwort ”Bayern” wurden sogar nur drei Nachrichten gefunden. Sie handeln von der Championsleague. Man sieht hieran, dass das Datenset hinsichtlich Sportthemen nicht umfangreich genug ist. Eine Evaluation dieses Graphen war so nicht möglich.

Für den DAX-Graphen wurden die besten aller Ergebnisse erzielt. Anscheinend funktioniert die Suche nach passenden Events für allgemeine Graphen besser. Ein Grund hierfür ist, dass die vom Nutzer gegebenen Ressourcen sehr allgemein sind. Weitere Ressourcen, die durch Methoden gefunden werden, sind meistens spezieller. Da aber auch Events zu diesen spezielleren Events den allgemeinen Graphen erklären können, ist die Precision hoch. In diesem Falle können zum Beispiel Events zur Wirtschaftslage von Samsung oder Apple den allgemeinen DAX-Kurs erklären. Die besten Ergebnisse für den Graphen wurden mit der *InOut Methode - Ohne Filter* und der *Ressourcen aus Events*-Methode mit *Subject Broader Filter* erzielt. Die Verlinkungsstruktur für allgemeine Wirtschaftsressourcen scheint so gut zu sein, dass diese Ansätze sich im Falle solcher Graphen lohnen und kaum unnütze Ressourcen hinzukommen. Im Falle der *InOut-Methode* und der *Ressourcen aus Events*-Methode wurde jedoch durch den Subject Broader Filter zu viel herausgefiltert, nämlich auch viele Events, die zum Thema passten. Das steht im Gegensatz zu den anderen Graphen, wo der Filter immer zu wenig filtert. Auch der *UClassify Filter* konnte beim DAX-Graphen am ehesten überzeugen, hier wurden viele News richtig eingestuft. Eine Erklärung hierfür ist, dass die Wirtschaftsnachrichten alle sehr generisch aufgebaut sind und bestimmte Begriffe wie ”increasing” oder ”sales” immer wieder im selben Zusammenhang genutzt werden.

Wie wir sehen, konnte sich keine Methode durchsetzen. Für jeden Graphen war eine andere Methode die beste, keine konnte für alle Graphen gute Ergebnisse erzielen. Die *Ressourcen aus Events*-Methode fand generell zu viele unnütze Events, ebenso die *InOut-Methode*. Die *Order by Relevance*-Methoden fanden meistens nur sehr spezielle Ressourcen, die im Datenset kaum genutzt wurden, hier stieg der Recall dann zu wenig an. Die besten Ergebnisse hinsichtlich der Precision wurden fast immer mit *Simple Methode* erzielt. Es war also nicht möglich, zusätzliche Ressourcen zur Suche nach Events mit einzubeziehen und dabei unnötige Events zu vermeiden. Die Filter funktionierten teils sehr gut, teils eher

schlecht. Der *Subject Broader Filter* filterte für den Apple- und Obama-Graphen zu wenige Events heraus. Allerdings wurde keines der Events herausgefiltert, die zum Thema passen, was sehr positiv ist. Beim DAX-Graphen wurden zwar mehr Events herausgefiltert, davon aber auch einige, die zum Thema passt.

Es ist allgemein sehr schwer, für alle Themengebiete eine Regel für die Struktur der Kategorien zu finden, die zu zufriedenstellenden Ergebnissen führt. Für den DAX-Graphen wäre eine `broader`-Stufe mehr besser gewesen, für den Apple- und Obama-Graphen eine `broader`-Stufe weniger. Der *UClassify Filter* konnte nur bedingt überzeugen. Einige Events wurden schlicht falsch eingestuft, in anderen Fällen war die Regel, nach der gefiltert wird, nicht gut. In einigen Fällen waren Events zum Beispiel nicht negativ genug, um herausgefiltert zu werden. Die Kombination beider Filter war im Falle der *InOut*- und der *Ressourcen aus Events*-Methode möglich. Die Kombination lieferte bessere Werte als die Filter im Einzelnen. Vor allem für den Obama- und den Apple-Graphen wurden so deutlich bessere Ergebnisse erzielt als mit einem einzelnen Filter. Das liegt daran, dass der *Subject Broader Filter* keines der Events herausfilterte, die zum Thema passten. Hierdurch wurde der Recall und die Anzahl an unnützen Events gesenkt, daraufhin konnte der *UClassify Filter* dann nochmals den Recall senken und die Precision steigern.

Graph	Passt und erklärt (absolut)	Passt und erklärt (prozentual)	Precision (prozentual)	Recall	Insgesamt
Apple	8 (RaE, IO)	0,5 (S_U)	1 (S)	226 (RaE)	S_U
Obama	30 (RaE)	0,41 (S_U)	0,82 (S_U)	985 (RaE)	IOO
DAX	36 (RaE)	1 (2LO_U)	1 (2LO, IOO)	490 (RaE)	RaE_SB

Tabelle 7.1: Die besten Methode nach Kategorie

Die Tabelle zeigt die besten Methoden in den aufgeführten Kategorien. Die Kategorie ”Insgesamt” ist eine Abwägung zwischen Recall, der prozentualen Precision und der Anzahl an Events, die zum Thema passen. Eine Methode wäre zum Beispiel insgesamt betrachtet schlecht, wenn sie zwar die meisten Events bietet, die zum Thema passen, aber gleichzeitig einen unglaublich hohen Recall hat und daher die prozentuale Precision sehr niedrig ist. Die Abkürzungen sind wie folgt:
S = Simple Methode
IO = InOut Methode
RaE - Ressourcen aus Events Methode
IOO = InOut Methode - Ordered by Relevance
2LO = Über zwei Links verwandte Ressourcen - Ordered by Relevance
_U = mit UClassify Filter
_SB = mit Subject Broader Filter

8 Ausblick

Wie wir gesehen haben, ist die automatische nicht-mathematische Analyse von Graphen noch weitgehend unerforscht. Diese Bachelorarbeit konnte weitere erste Schritte auf diesem Gebiet machen, gelöst ist das Problem aber nicht. Keiner der Ansätze konnte vollends für jeden Graphen überzeugen. Für allgemein gehaltene Graphen, die aus dem Wirtschaftsbereich kommen, konnten wir bereits gute Ergebnisse erzielen. Für Graphen aus der Politik waren die Erfolge eher weniger befriedigend. Über Graphen aus dem Sportbereich konnte keine Aussage getroffen werden.

Das erste Problem ist, dass das verwendete Datenset nicht umfangreich genug ist. Zwar finden sich zahlreiche Events in den Jahren 2010 bis 2012, die Jahre davor sind aber nur sehr spärlich abgedeckt. Dadurch lassen sich nur Graphen aus der jüngsten Zeit zufriedenstellend analysieren. Doch auch in den Jahren 2010 bis 2012 sind längst nicht genug Events vorhanden. Vor allem aus den Bereichen ”Sport” oder ”Weltgeschehen” beziehungsweise ”Klatsch und Tratsch” sind zu wenige bis gar keine Events vorhanden. Das ist auch der Grund, warum zu einem unserer verwendeten Graphen, dem Bayern-München-Punktegraph, kein einziges Event gefunden wurde. Das Datenset müsste also umfangreicher und vor allem von den Themen her breiter gefächert sein. Da wir eine lokale Kopie des Datensets verwendeten, gab es zudem keine Updates mit neuen Events, ab Ende 2012 waren also keine Events mehr im Datenset. Ein erster Schritt wäre, auf den Endpoint des Datensets Zugriff zu erhalten und diesen zu verwenden.

Man kann sich natürlich auch die theoretische Frage stellen, was wäre, wenn man auf ein Datenset Zugriff hätte, das alle Nachrichten der Welt enthält und darüber hinaus perfekt getaggt ist. Dass also zu jeder Nachricht genau die Ressourcen unter /textttinvolved stehen, die mit dieser Nachricht zu tun haben. In diesem Fall würde schon der naive Ansatz ausreichen, bei der Suche nach Events zur Analyse des Graphen nur die vom Nutzer verwendeten Ressourcen zu verwenden. Allerdings würden auch in diesem Fall Nachrichten, die nur indirekt Einfluss auf den Graphen haben, verloren gehen. Ein Beispiel wäre hier die Nachricht zur allgemein guten Wirtschaftslage, die auch eine Erklärung für den Anstieg eines speziellen Aktienkurses ist.

Das zweite Problem ist, dass allgemein die Fragestellung nach verwandten Ressourcen im Semantic Web noch nicht gelöst ist. Wie wir in dieser Bachelorarbeit gesehen haben, sind eher einfache Ansätze keine zufriedenstellende Lösung. Alle eingehenden und ausgehenden Links einer Ressource zu verwenden, führt zum Beispiel zu übermäßig vielen Ressourcen, die nicht mehr zum Thema passen. Ansätze, die über die Struktur des `dcterms:subject`-Prädikats und die `broader` und `broader of` Struktur gehen, liefern zumindest für Wirtschaftsgraphen gute Ergebnisse. Es werden allerdings tendentiell zu wenige Ressourcen gefunden und auch hier finden sich themenfremde Ressourcen.

Die Frage, welche Ressourcen themenverwandt sind, ist darüber hinaus allgemein sehr schwer zu beantworten. Nehmen wir als Beispiel die Ressource Darmstadt. Im Hinblick auf ”Städte mit einer Universität” wäre zum Beispiel ”Frankfurt” eine verwandte Ressource. Im Hinblick auf ”Städte mit unter 500.000 Einwohnern” wäre Frankfurt dagegen nicht verwandt. Der Nutzer müsste also hier von Hand angeben, im Hinblick auf welches Thema verwandte Ressourcen gefunden werden sollen.

Das dritte Problem ist es, Nachrichten wieder herauszufiltern, die zwar zum Thema des Graphen passen, aber nicht den Verlauf erklären. Die Nachricht ”Apple vermeldet Rekordgewinne” passt zwar zu einem Apple-Aktienkursgraphen, wenn diese Nachricht aber während eines negativen Trends gefunden wird, kann sie den Verlauf nicht erklären. Mit Klassifizierern wie *UClassify* gibt es bereits Programme, die versuchen, Texte einzustufen. Allerdings funktioniert die Klassifizierung teils noch nicht, eine Nachricht über mehrere Tote bei einem Anschlag etwa wurde als positiv eingeschätzt. Dies liegt nicht daran, dass *UClassify* im Speziellen schlecht ist, sondern dass das Problem der Klassifizierung noch nicht zufriedenstellend gelöst wurde. Ein Paper aus dem Jahr 2007 verglich verschiedene Textklassifizierer anhand eines Sets von Nachrichtenheadlines [8]. Auch die besten Klassifizierer konnten nur um die 60 Prozent Precision aufweisen. Hier gibt es also allgemein noch Weiterentwicklungsbedarf.

Ein erhebliches Problem beim Einschätzen einer Nachricht ist, dass die Bewertung immer auch im Auge des Betrachters beziehungsweise des Betroffenen liegt. Die Nachricht ”Apple vermeldet Verluste” ist für Apple zum Beispiel eine negative Nachricht. Für einen direkten Konkurrenten am Smartphonemarkt wie Samsung ist dies jedoch eine gute Nachricht. Für einen Aktienkursgraphen von Samsung könnte diese Nachricht also den Anstieg des Graphen erklären. Programme, die den Betrachter miteinbeziehen, gibt es bisher nicht. Solch ein Programm müsste wissen, wer der Betrachter ist, und müsste weiterhin wissen, wer von positiven beziehungsweise negativen Nachrichten über diesen Betrachter profitiert. Das ist

bisher nur von Hand zu lösen.

In allen drei wichtigen Feldern, auf denen diese Bachelorarbeit beruht, gibt es also noch sehr viel Entwicklungsbedarf. Das Datenset muss umfangreicher werden, das Problem der themenverwandten Ressourcen muss stärker erforscht werden und die Textklassifikation, die auch den Betrachter miteinbezieht, muss angegangen werden.

Danksagung

Danke an Heiko Paulheim für die umfassende und gute Betreuung dieser Thesis.
Danke an Gerd Holthausen und Luise Holthausen-Hiss für das Korrekturlesen.

References

[1] Berners-Lee et al. (2001): The Semantic Web. In: Scientific American, Mai 2001
[2] http://dbpedia.org/About, last visited 28.02.2013
[3] Berners-Lee, T. (2006). Linked Data. http://www.w3.org/DesignIssues/LinkedData.html.
[3] Harris, Seaborne, Prud'hommeaux (2013). SPARQL 1.1 Query Language. http://www.w3.org/TR/sparql11-query/
[4] Klyne, G. and Carroll, J. J. (2004). Resource description framework (rdf): Concepts and abstract syntax. http://www.w3.org/TR/rdf-concepts/
[5] W3C. http://www.w3.org/Consortium/mission.html . last visited 24.03.2013
[6] Hienert, Wegener, Paulheim. Automatic Classification and Relationship Extraction for Multi-Lingual and Multi-Granular Events from Wikipedia. 2012
[7] Hienert, Luciano. Extraction of Historical Events from Wikipedia. 2012
[8] Strapparava, Mihalcea. SemEval-2007 Task 14: Affective Text. 2007
[9] Berners-Lee, T. Uniform Resource Identifier (URI): Generic Syntax. http://tools.ietf.org/html/rfc3986. 2005
[10] Dan Brickley, Libby Miller. FOAF Vocabulary Specification. http://xmlns.com/foaf/spec/ . 2010
[11] Robert D. Edwards,John Magee,W.H.C. Bassetti. Technical Analysis of Stock Trends, Ninth Edition. 2007
[12] Peter Hamby. Analysis: Why Romney lost. http://www.cnn.com/2012/11/07/politics/why-romney-lost . last visited 16.4.2013
[13] Karl Erik Wrneryd. Stock-market Psychology: How People Value and Trade Stocks. 2001
[14] UClassify. http://uclassify.com/ . last visited 15.4.2013