Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

Web-basierte Methoden zur Untersuchung von Affiliation-Angaben wissenschaftlicher Papiere

Bachelorarbeit

Leipzig, Juni, 2010

vorgelegt von:

Xia, Chaohui

Studiengang: BSc. Informatik

Betreuender Hochschullehrer: Prof. Dr. Erhard Rahm

Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei Prof. Dr. Erhard Rahm für die Überlassung des Themas und bei Herrn David Aumüller für die Betreuung der Bachelorarbeit bedanken.

Mein besonderer Dank gilt Herrn David Aumüller für seine Hilfsbereitschaft, freundliche Unterstützung und seine vielen kritischen Hinweise, die mir bei der Verbesserung der Analyseergebnisse geholfen haben. Ich erhielt von ihm viele Anregungen für meine Arbeit und habe immer sein offenes Ohr für Rückfragen bei der Bewältigung diverser Probleme gefunden.

Zum Schluss ein Dankeschön an die Universität Leipzig, die den Mysql Server und die Homepage zur Verfügung stellten.

Chaohui Xia

2010-08

Inhaltsverzeichnis

I. Einführung

1.1 Motivation

Bei der zunehmenden Anzahl von Papers spielt die Affiliationsanalyse eine größer werdende Rolle, man erstrebt zusätzlich zur bibliografischen Analyse, um weitere nützliche Information zu erhalten. So wird z.B. die Affiliation nach Instituten, Countrys, Regionen, Citys und Koordinaten, die aus der Originalaffiliation nicht direkt bekannt sind, analysiert, anschließend kann mit diesen Informationen weiter gearbeitet werden. Man kann die Affiliation mit dem gefundenen Ergebnis kontrollieren und verteilen. Weltweit existieren zahlreiche semantische Analysewerkzeuge. Hier soll die Rede von vier grundsätzlichen Analysemethoden sein und am Ende wird die Auswertung nach einer einzigen Metrik stehen.

Durch diesen gefundenen Ergebnisse können wir viele Hinweise gewinnen, z.B. den das Trends der Papers innerhalb von zehn Jahren errechnen und dann weiteres mit Hilfe von GoogleMaps durch die MarkerCluster Technik anzeigen. Auch können wir die Papers nach verschiedenen Ansätzen verlinken. Es wird möglich eine Reihe von Beziehungen definiert, um ein Netz aufzubauen. Dieses kann schnell durchgesehen und es können mächtige Suchstrategien ausgeführt werden. Diese Idee ist unser heutiges Thema und auch unter dem Begriff des Semantik Web bekannt.

In den zahlreichen bibliografischen Datenbanken existieren keine tauglichen Standards hierfür, das erschwert es, die Anfragen über die vorhandenen Inhalte auszuführen. So enthält zum Beispiel die Google Scholar viele inkonsistente Angaben im Attribute ‚Venue', das z.B. eine Konferenz oder ein Journal bezeichnet. Hier sind viele alternative Formate enthalten, dies schwächt die Qualität der Anfrage erheblich. Eine typische Methode zur Verbesserung der Suchergebnisse ist die Einbindung einer Authority-Datei, d.h. eines Katalogs oder Synonymwörterbuchs über verschiedene Namen und Abkürzungen einer Konferenz oder eines Journals. Die Authority-Datei

verweist somit auf einen eindeutigen String für das bibliografische Feld, d.h. es wird ein bibliografischer Index erzeugt. French, Powell und Schulman versuchen in [14] mithilfe des Levenshtein-Ähnlichkeitsmaß die Ähnlichkeit bzw. den Abstand zwischen zwei Strings zu berechnen und so einen solchen Katalog für Autor-Affiliation-Angaben zu erstellen. Neuere Ansätze versuchen dazu das Wissen von Websuchmaschinen auszunutzen. Pereira, D. et al. [13] werten die Snippets von Suchanfrageergebnissen zur Erstellung einer Authority-Datei von Venue-Varianten aus. Erscheint z.B. die Abkürzung eines Venues in der Suche nach dem Langnamen und umgekehrt, gehören die zwei Strings zusammen.

Das Schema der Erstellung der Authoritydatei über die Venue-Information stellt die folgende Abbildung 1.1 dar, am Ende werden alle Venues nach Name und Venue Type geclustert.

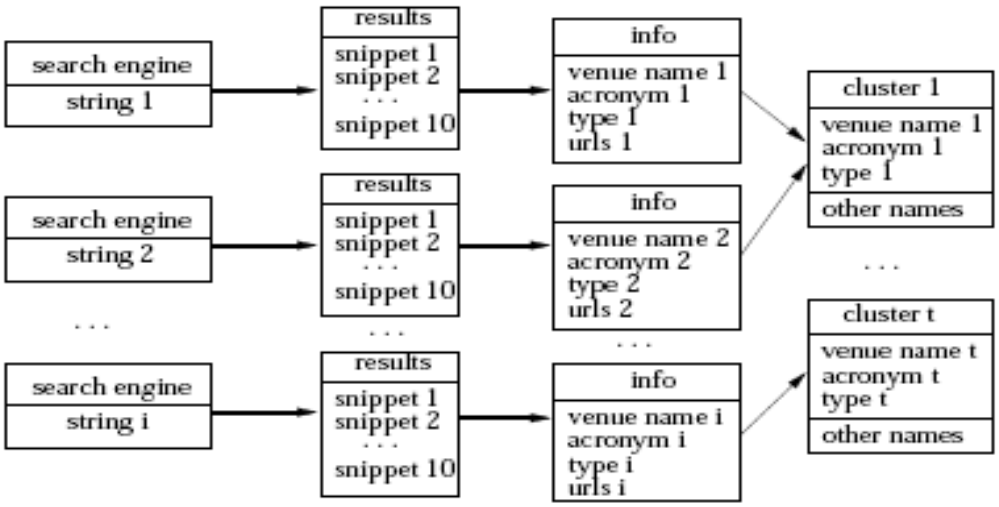


Abbildung 1.1 Schema für Erstellung der Authority-Datei von ‚Venue'[13]

D. Aumüller und E. Rahm[8] betrachten ebenso die Ergebnisse von Webanfragen zur Erstellung einer Affiliationdatenbank. Jeder Affiliation-String wird als Query an eine Web-Suchmaschine (z.B. Google) angefragt, falls die Suchergebnisse verschiedener Strings ähnlich ausfallen (z.B. die erste URL zweier Anfragen sind identisch), wird angenommen, dass auch die Strings der Anfrage „ähnlich" sind (z.B. „MIT" und „Massachusetts Institute of Technology") und diese als das selbe Institut betrachtet werden können.

In vorliegender Bachelorarbeit werden die Affiliation-Strings durch verschiedene spezielle Web-Services analysiert und versucht, so von den heterogenen Schreibweisen auf eine homogene Variante zu gelangen. Drei web-basierte und ein lokaler Ansatz werden untersucht: OpenCalais, Google Ajax Search API, und Google Search auf Wikipedia (Web-Wiki), und eine Kopie der DBPedia-Datenbank, die nur die relevanten Wikipedia-Infoboxen in Tabellen gespeichert enthält. Ziel ist es z.B. dass wir sowohl bei der Eingabe der Affiliation wie ‚Universität Leipzig' als auch ‚Uni Leipzig' die gleiche Institutsangabe, z.B. „University of Leipzig" mit Region Saxony und Land Germany, erhalten. Zum Schluss werden wir die Qualität einer jeden Methode bewerten und die Papers mit dem zugehörigen Ergebnis darstellen.

1.2 Aufgabenstellung

Das Ziel dieser Bachelorarbeit ist es, durch Eingabe der Affiliation das Institut und den Ort zu erkennen. Das Ergebnis wird nach den verschiedenen Analysemethoden in einer Datenbank gespeichert und kann danach in der erwünschten Form angezeigt (z.B. Google Maps, Google Chart) oder mit den richtigen Informationen verglichen bzw. ausgewertet werden (z.B. Precision Recall & F-Measure Metrik).
Den ganz einfachen Datenfluß kann man wie in Abbildung 1.2 zeichnerisch darstellen. Die Filterung der Ergebnisse besteht im Auslesen von wichtigen Informationen und der Mehrheitsentscheidung, wenn sich mehr als ein Ergebnis ergibt:

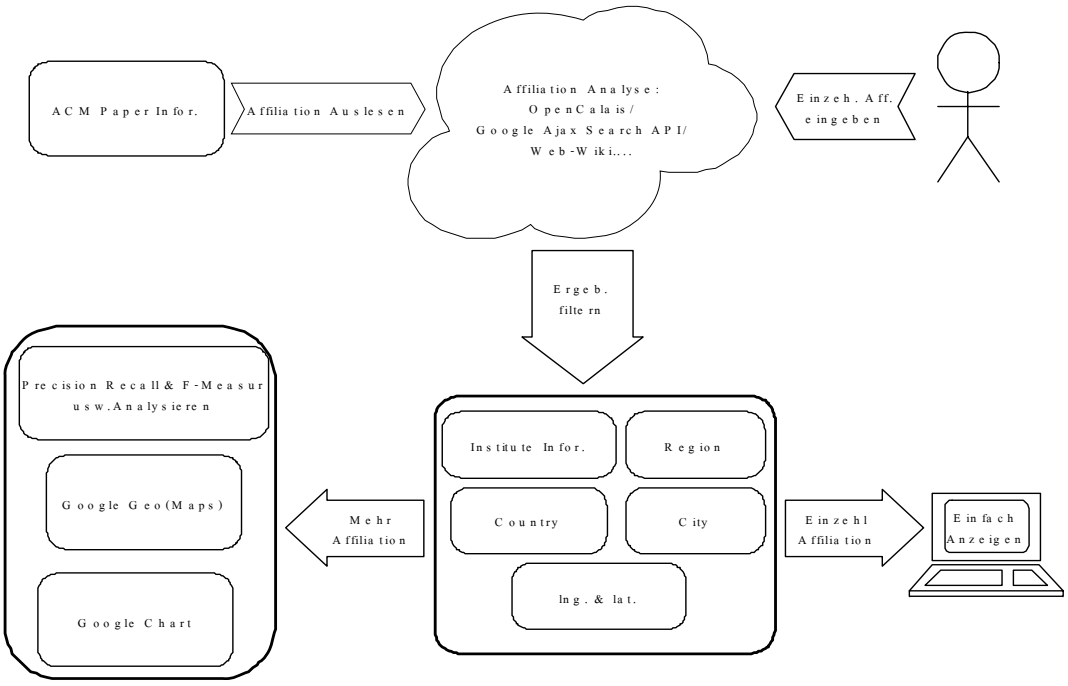


Abbildung 1.2 Systemstrukturen

1.3 Gliederung der Arbeit

Die Bachelorarbeit gliedert sich wie folgt:

In Kapital II werden nach einer Einführung die verschiedenen Analysemethoden und dazu die Qualitätsbewertung und Anzeigetechnik erläutert. Es bieten sich vier grundsätzliche Analysemethoden an (2.1 OpenCalais, 2.2 Google Ajax Search API, 2.3 Google Search auf Wikipedi, 2.4 DB-Pedia), das zurückgelieferte Ergebnis wird mit einer manuell eingegebenen richtigen Information über Institut und Ort in Bezug auf Precision, Recall und F-Measure bewertet. Es können auch eine bestimmte Menge von Affiliationen im Google Maps angezeigt werden (2.6 Google Geo) oder die Anzahl der Papers nach Land und Jahr analysiert werden (2.6 Google Chart).

Das Kapital III beschäftigt sich mir der Vorgehensweise der Analyse. Es können zahlreiche Order einer Affiliation vom Autor eingegeben werden. Dies wird anhand eines Beispieles gezeigt und der Algorithmus konkret erklärt.

Im Kapital IV wird auf die Implementierung eingegangen, es wird erklärt, wie das System installiert wird und wie die Datenbankstrukturen bzw. die Attribute der Tabellen und ihre Beziehungen zueinander (Foreign Key) aussehen.
Zum Schluss werden die Probleme nochmals benannt und es wird ein Ausblick auf künftig anstehende Arbeiten genommen.

II. Grundlagen

2.1 OpenCalais

Einen in der Nutzung des stetig wachsenden Beitrag für das „Web of Data“ leistet OpenCalais, das als kostenloser Web-Service unstrukturierte Texte analysiert. Es extrahiert daraus vorkommenden Personen, Organisationen, Orte, Fakten und Ereignisse und analysiert die Beziehungen dieser Entitäten zueinander[7].

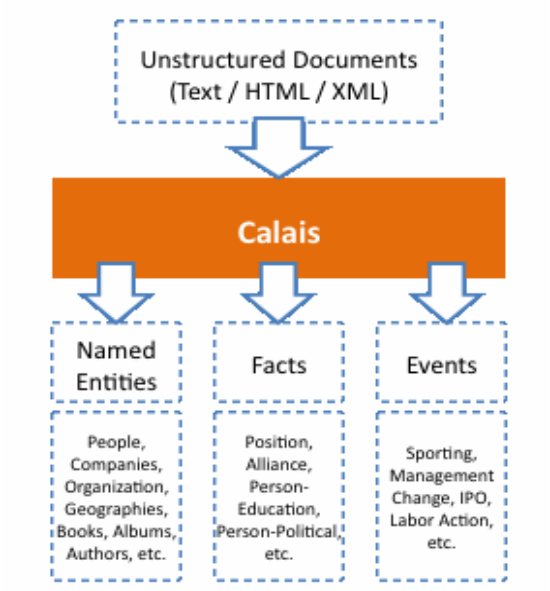


Abbildung 2.1 Calais identifiziert Entitäten, Fakten und Ereignisse[1]

Zur Analyse sendet man den unstrukturierten, natürlichsprachigen Content als folgendes Format:

TEXT/HTML: Es braucht keine Konvention von Content und es werden unrelevante Contents sowie HTML Tags und –Scripte automatisch entfernt. Entitäten und Ereignis werden mittels relativer Order gereinigt und der Text extrahiert

TEXT/HTMLRAW: mächtiger als TEXT/HTML. Man kann eine irreale HTML eingeben, dann versucht OpenCalais, die Quelle real anzusteuern. Dann extrahiert es wie TEXT/HTML.

TEXT/XML: Mit OpenCalais können auch die XML Daten geparst werden. Es unterstützt den NewsML Standard. OpenCalais wird durch folgende XML Knoten den reinen TEXT extrahieren.

Document Section	Supported XML Tag Names
Document Title	TITLE, HEADLINE, HEADER
Document Body	BODY, DESCRIPTION, CONTENT
Document Date	DATE, DATETIME, DATEANDTIME, PUBDATE

Abbildung 2.2 XML zu reinen Text[3]

TEXT/RAW: Entitäten und Ereignisse werden direkt extrahiert. Wenn die eingegebenen Kontexte schon reine Text sind, bedarf es keiner gesonderten Behandlung.

Bemerkung: Das TEXT/TXT Format ist nicht mehr unterstützt, wenn keine Format angegeben ist, werden die Eingabe- order Ausgabeformate als TEXT/RAW betrachtet.

Nachdem eines von fünf Formatdaten oben genannten gesendet wurde, erhält man semantische Metadaten als RDF, Microformats, Simple Format oder JSON zurück.

Auf der Homepage von OpenCalais kann man diese mit dem Dokument Viewer [2] einfach ausprobieren. Dort kann einfach ein String eingegeben werden, dann werden automatisch viele Information bei Zerlegung des Strings erkennbar.

2.2 Google Ajax Search API

Die Google Ajax Search API bietet viele API Funktionen an, in dieser Arbeit wird die API von Lokal-Search, Web-Search und Image-Search genutzt.

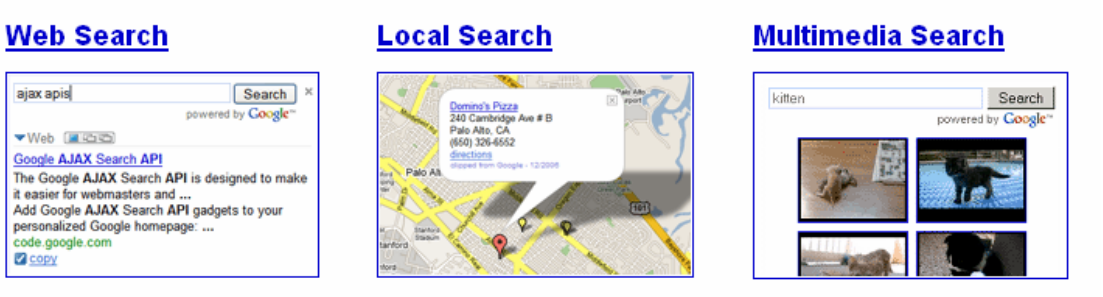


Abbildung 2.3 Drei mögliche Google Ajax Search API [5]

Bei der einfachen Anfrage als Ortsanfrage kann man die folgende URL besuchen:

Zu Beachtung: Wie in folgender URL gezeigt wird, ist der Parameter q der Anfragestring und der Parameter key der API Key, die von der Homepage von Google Ajax Search beantragt werden können.

http://ajax.googleapis.com/ajax/services/search/local?v=1.0&q=...&key=...

Analog sieht die Anfrage von Websearch so aus:

http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=...&key=...

Und für die Imagesuche:

http://ajax.googleapis.com/ajax/services/search/images?v=1.0&q=...&key=...

Es werden immer JSON-Daten zurückgeliefert.

2.3 Google Search auf Wikipedia

In der Methode von Google Search auf Wikipedia (kurz Web-Wiki) gibt es zwei Teile, zuerst mals wird durch die Web Google Ajax Search API analysiert, um die Webseite von Wiki zu finden, anschließend wird der Code von der gefundenen Webseite

herunterladen und es werden die relevanten Daten auslesen. Wenn wir die Affiliation als , The University of Texas at Arlington' eingeben:
http://ajax.googleapis.com/ajax/services/search/web?v=1.0&key=...&q=
site:wikipedia.org/wiki/ The University of Texas at Arlington
Die erhaltene englische Seite von Wiki (Nach der Bearbeitung) lautet dann:
http://en.wikipedia.org/w/index.php?title=University_of_Texas_at_Arlington
Man kann auch diese Webseite einfach direkt besuchen und klickt dann das Botton „Bearbeiten". Im Wikipedia gibt es sehr viele nützliche Informationen, leider wird keine Koordinateninformation mitgeliefert.

2.4 DB-Pedia

Die letzte Methode heißt DBPedia. Sie versucht, die Informationen von Wikipedia nach bestimmten Beziehungen zu erstellen und im Web wieder zu aktivieren. Die meisten Datenquellen werden als RDF Datei, die auf der Homepage von DBPedia stehen[6], gespeichert. Hier wird in der Infobox über University, Company, Organisation usw. nachgeschlagen, ob durch DBPedia die Informationen vom Institut und dem Ort erlangt werden können. Die relevanten Informationen der Infobox wurden aus DBPedia extrahiert und gesondert in der Datenbank gespeichert. Die Quellendaten der DBPedia kann man von der Homepage der DBPedia [6] herunterladen. Es existieren viele Sprachversionen. Wir entscheiden uns hier für die Englischversion an. Es werden anschließend alle Institute mit ihren Orten in der Datenbank gespeichert und dann analysiert man durch SQL Anfrage, ob einige Citys, Regionen oder Countrys usw. angepasst werden können.

Dataset	en	de	fr	pl	ja	it	nl	es	pt	ru	sv	zh
DBpedia Ontology (preview)	owl	--	--	--	--	--	--	--	--	--	--	--
Ontology Infobox Types (preview)	nt nq	--	--	--	--	--	--	--	--	--	--	--
Ontology Infobox Properties (preview)	nt nq	--	--	--	--	--	--	--	--	--	--	--
Ontology Infobox Properties (Specific) (preview)	nt nq	--	--	--	--	--	--	--	--	--	--	--
Titles (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq
Short Abstracts (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq
Extended Abstracts (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq
Images (preview)	nt nq	--	--	--	--	--	--	--	--	--	--	--
Links to Wikipedia Article (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq
Articles Categories (preview)	nt nq	--	--	--	--	--	--	--	--	--	--	--
External Links (preview)	nt nq	--	--	--	--	--	--	--	--	--	--	--
Raw Infobox Properties (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq
Raw Infobox Property Definitions (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq
Homepages (preview)	nt nq	nt nq	nt nq	--	--	--	--	--	--	--	--	--
Geographic Coordinates (preview)	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq	nt nq

Abbildung 2.4 Downloadseite von DBPedia [6]

2.5 Precision, Recall und F-Measure

Um das Ergebnis von verschiedene Methoden zu evaluieren, benötigt man die Affiliation mit ihren entsprechenden korrekten Informationen über Institute, Orte und Koordinaten. Angenommen wir haben alle eingegebenen Affiliationen mit den korrekten Instituten und Orten (oder machmal auch mit den entsprechenden Koordinaten) in der Datenbank gespeichert. Eine klassische Metrik berechnet:

Precision = |true positives| / (|true positives| + |false positives|)

Recall = |true positives| / |real correspondences|

F-Measure = 2 * Precision * Recall / (Precision + Recall) (Quelle : [8])

Precision (Genauigkeit) : Verhältnis der gefundenen relevanten Dokumente zur Gesamtmenge gefundener Dokumente.

Recall (Trefferquote) : Verhältnis gefundener relevanter Dokumente zur Gesamtmenge relevanter Dokumente.

F-Measure: Harmonisches Mittel von Precision und Recall.[9]

True positives: Anzahl der gefundenen Affiliationen, bei denen die richtige Information erhalten werden.

False positives: Anzahl der gefundenen Affiliationen, die falsche Information enthalten.

Real correspondences : Anzahl aller eingegebenen Affiliationen.

Zum Beispiel haben wir 10 Affiliationen eingegeben. 8 Affiliationen haben die Orten gefunden und innerhalb dieser gibt es 6 Affiliationen mit den richtigen Orten und 2 Affiliationen mit falscher Orten, dann folgt daraus:

True positives = 6 , **False positives** = 2 , **Real correspondences** = 10

Precision = 6 / 8 = 0.75

Recall = 6 / 10 = 0.6

F-measure = (2 * 0.75 * 0.6) / (0.75 + 0.6) = 0.6667

Bei der Vergrößerung der Recall wird die Precision fast linear gesenkt. Der typische Verlauf eines Recall-Precision-Graphen sieht so aus:

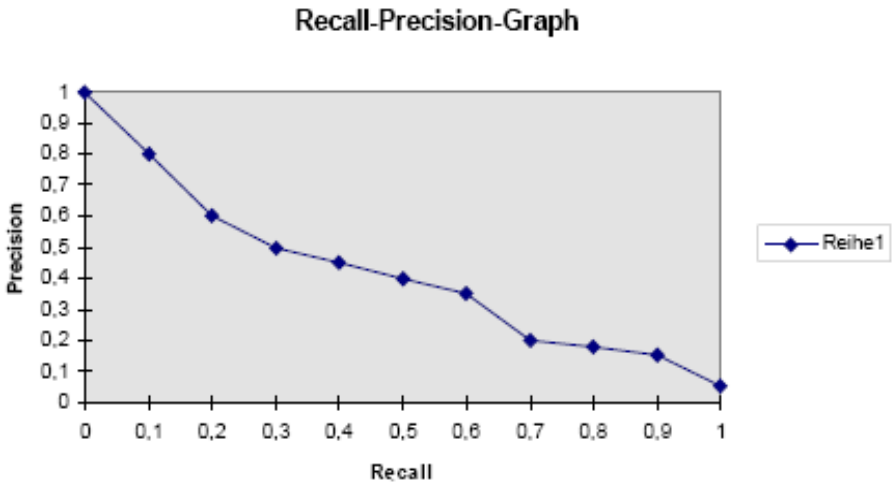


Abbildung 2.5 Typischer Verlauf eines Recall-Precision-Graphen [9]

2.6 Google Geo & Google Chart

Wenn viele Marker auf einer Google Map angezeigt werden, sinkt die Performance des Browsers stark. Um dieses Problem zu lösen, kann man einige Marker, die einander ganz nahe gelegen sind, clustern und die Anzahl der Marker ins Zentrum stellen. Die Idee vom MarkerCluster zeigt die Abbildung 2.6, im Bild links werden alle Koordinaten diskret angezeigt, rechts im Bild wird die Google MarkCluster Technik genutzt:

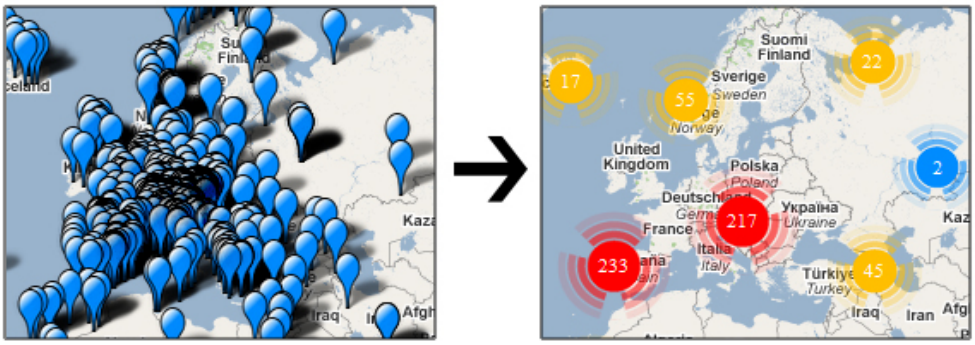


Abbildung 2.6 MarkerCluster Technik[11]

Die MarkerCluster hängen natürlich von Auflösung des Bildes ab. Wenn das Bild maximiert wird, werden die Durchmesser der Clusterkreise verkleinert bzw. die Marken werden auf dem Map verstreurt. Von Google Chart wird eine Webapplikation von Google angeboten. Mit ihr können die evaluierten Daten sehr deutliche und effektiv dargestellt werden. Es gibt zwei Varianten von Google Chart, die erste ist die Image Chart und die zweite die Interactive Chart:

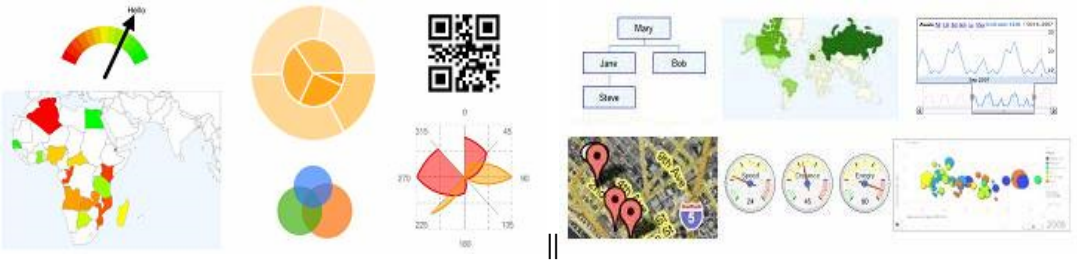


Abbildung 2.7 Zwei Arten des Google-Chart [10]

Die Image Charts sind statische Bilder, die sehr schnell erzeugt werden können (linkes Bild von Abbildung 2.7). Bei den Interactive Charts wird vom Server einem Ereignis zurückgeliefert und der Nutzer kann dies behandeln als seinen eigenen Code (rechtes Bild von Abbildung 2.7).

Im diesem Projekt werden die Interactive Charts genutzt. Es wird die Anzahl der Papers nach Land und Jahr berechnet und im Histogramm, dem Scattplott und der Parabell Linie angezeigt. Einem einfachen Beispiel wird am folgenden Kapital konkret abgearbeitet.

III. Ablauf der Affiliationanalyse

3.1 Interaktive Affiliationsanalyse

Die Datenquelle stammt aus dem ACM Paper, die Information enthält den Paper Link, den Autor, das Jahr, und die Affiliation. Die Datenquelle im csv Format sieht wie folgt aus:

“342009”;”sigmod”,”2000”,”research”;”335372”;”1”;”Jiawei Han”;”School of Computing Science, Simon Fraser University”
“342009” ;”sigmod”,”2000”,”research”;”335372”;”2”;”Jian Pei”;”School of Computing Science, Simon Fraser University”
“342009” ;”sigmod”,”2000”,”research”;”335372”;”3”;”Yiwen Yin”;”School of Computing Science, Simon Fraser University”...

Abbildung 3.1 csv-Datei als Quelle

Es gibt insgesamt 3643 Affiliationen und darin sind ca. 1005 unterschiedliche enthalten. Bei jedem Datensatz wird so eingeordet: Venid, venue, year, track, acmid, rank, name, affiliation. Z.B. wird der erste Datensatz wie folgt extrahiert (venue als sigmod): venid = 342009, venue = sigmod, year = 2000, track = research, acmid=335372, rank = 1, name = Jiawei Han, Affiliation = School of Computing Science,

Simon Fraser University, das Link vom Paper wird mit folgender Formel errechnet:

http://portal.acm.org/citation.cfm?id=venid.acmid

wie bei obigen Beispiel:

http://portal.acm.org/citation.cfm?id=342009.335372

Es gibt insgesamt zwei Schritte der Analyse. Im ersten Schritt speichert man die Quelle in der Datenbank, dabei kann man die CSV Datei importieren oder auch manuell in der Tabelle ‚quelle' verarbeiten. Im zweiten Schritt kann man eine gültige Quelle-ID auswählen und anschließend analysieren.

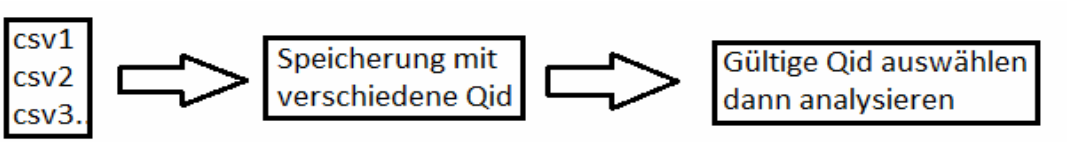


Abbildung 3.2 Analysevorgehensweise

csv1, csv2, csv3 sind die Quell-Dateien. Bei der Auswahl der gültigen Qid wird das System überprüfen, ob die ausgewählte Qid schon vorher ausprobiert wurde. Bei der Analyse wird das System wieder nur die Affiliation, die vorher noch nicht analysiert wurde, analysieren. Es bieten sich acht verschiedene Analysemethoden an:

1). OpenCalais

2). Google Ajax Search API

3). Google Search auf Wikipedia (Web-Wiki)

4). DBPedia

5). OpenCalais & Google Ajax Search API

6). OpenCalais & Google Ajax Search API & Web-Wiki

7). OpenCalais & Google Ajax Search API & Web-Wiki & DBPedia

8). OpenCalais + Google Ajax Search API + Web-Wiki + DBPedia

Abbildung 3.3 Acht Analysemethoden

Wir können die obigen 8 Analysemethoden in 4 Schichten darstellen, sie lauten:

1. Einzelne Analysemethode
2. OpenCalais & Google AJAX Search API
3. OpenCalais & Google Ajax Search API & Web-Wiki
4. OpenCalais & Google Ajax Search API & Web-Wiki& DBPedia

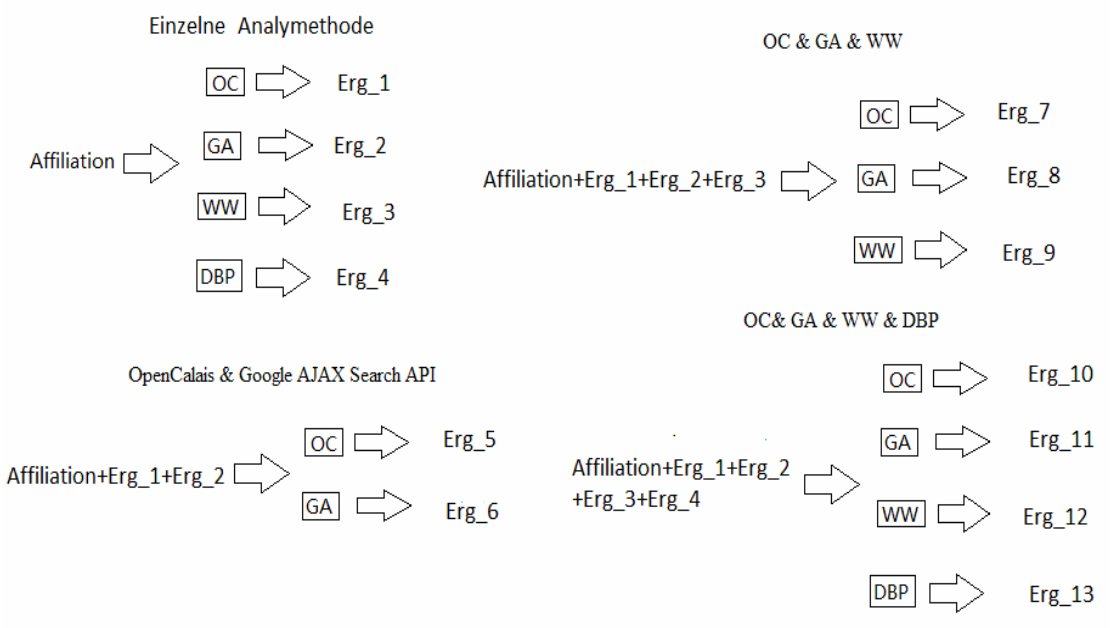


Abbildung 3.4 Schichtung der Analysemethoden

Damit bekommt am als finale Ergebnis der Analysemethoden das Folgende:

Nur durch OpenCalais (OC): Erg_1

Nur durch Google Ajax Search API (GA): Erg_2

Nur durch Web-Wiki (WW): Erg_3

Nur durch DBPedia (DBP): Erg_4

OC + GA + WW+ DBP: { Erg_1, Erg_2, Erg_3, Erg_4 }

OC & GA: { Erg_1, Erg_2, Erg_5, Erg_6 }

OC & GA & WW: { Erg_1, Erg_2, Erg_3, Erg_5, Erg_6, Erg_7, Erg_8, Erg_9 }

OC & GA & WW & DBP: { Erg_1, Erg_2, Erg_3, Erg_4, Erg_5, Erg_6, Erg_7, Erg_8, Erg_9, Erg_10, Erg_11, Erg_12, Erg_13 }

Zubeachten: {Erg_1,....} bedeutet Mehrheitsentscheidung für das Ergebnis.

Ich werde die Vorgehenweise mit einem enfachen Beispiel erklären. Angenommen wir haben die Affiliation mit ‘Uni Leipzig’ eingegeben:

Nur durch OpenCalais (OC):

Bei der Methode für ‚Nur durch OpenCalais zu analysieren’ bekommen wir das Ergebnis als Erg_1:

City: Leipzig, Longitude: 12.38, Latitude: 51.34.

Nur durch Google Ajax Search API (GA):

Bei der Methode für ‚Nur durch Google Ajax Search API zu analysieren’ bekommen wir das Ergebnis als Erg_2:

Institute: Universität Leipzig, City: Leipzig, Country: Deutschland, Longitude: 12.39261, Latitude: 51.3283.

Nur durch Google Search auf Wikipedia (WW):

Bei der Methode für ‚Nur durch Google Search auf Wikipedia (Web-Wiki) zu analysieren’ bekommen wir das Ergebnis als Erg_3:

Institute: University of Leipzig, City: Leipzig, Region: Saxony, Country: Germany.

Nur durch DBPedia (DBP):

Bei der Methode für ‚Nur durch DBPedia analysieren’ bekommen wir das Ergebnis als Erg_4:

Institute: Thomasschule zu Leipzig, City: Leipzig, Region: Saxony, Country: Germany.

OpenCalais & Google Ajax Search API (OC & GA):

Die Eingabe ist ‚Uni Leipzig’ + Erg_1 + Erg_2 : ,

Uni Leipzig , , Universität Leipzig , Leipzig , Leipzig , , , Deutschland’.

Die Regel von ‚Uni Leipzig’ + Erg_1 + Erg_2 ist:

Affiliation , Institut (Erg_1) , Institut (Erg_2) , City (Erg_1) , City (Erg_2) , Region (Erg_1) , Region (Erg_2) , Country (Erg_1) , Country (Erg_2)

D.h. Die alle unter der Linie stehenden sind Erg_1 mit der Reihenfolge Institute, City (Leipzig), Region und Country' , dann reinigen wir die Strings:

‚Uni Leipzig, Universität Leipzig, Leipzig, Deutschland'

anschließende analysieren wir die Eingabe nach OpenCalais (Erg_5) und Google Ajax Search API(Erg_6):

Erg_5: City: Leipzig, Country: Deutschland, Latitude: 51.34, Longitude: 12.38.

Erg_6: Institute: University of Leipzig, Region: Sachsen, City: Leipzig, Country: Germany, Latitude: 51.34127, Longitude: 12.37983.

Das Ergebnis von OC & GA ist die Mehrheitsentscheidung von Erg_1, Erg_2, Erg_5 und Erg_6, der Anzahl der entsprechende Informationen sieht folgendermaßen aus:

City: Leipzig(4), Institute: Universität Leipzig(1), University of Leipzig(1), Country: Deutschland(2), Germany(1), Region: Sachsen(1).

Nach der Mehrheitsentscheidung sieht das finale Ergebnis wie folgt aus: (Bei gleiche Häufigkeit wählen wir den erste Wert.)

Institute: Universität Leipzig, City: Leipzig, Region: Sachsen, Country: Deutschland, Longitude:12.37983, Latitude: 51.34127.

OpenCalais & Google Ajax Search API & Google Search auf Wikipedia (OC & GA & WW):

Das Eingabe ist analog der OC & GA, es wird die Affiliation als ‚Uni Leipzig' + Erg_1, Erg_2 + Erg_3 eingegeben, dann versuchen wir wieder nach OpenCalais (Erg_7), Google Ajax Search API (Erg_8) und Webwiki (Erg_9), zu analysieren. Das finale Ergebnis von OpenCalais & Google Ajax Search API & Webwiki wird durch die Mehrheitsentscheidung von Erg_1, Erg_2, Erg_3, Erg_5, Ergs_6, Erg_8 und Erg_9 bestimmt:

Institute: Universität Leipzig, City: Leipzig, Region: Sachsen, Country: Deutschland, Longitude: 12.37983, Latitude: 51.34127.

OpenCalais & Google Ajax Search API & Google Search auf Wikipedia & DBPedia (OC & GA & WW & DBP):

Das Eingabe ist analog der OC & GA, es wird die Affiliation als ‚Uni Leipzig' + Erg_1 + Erg_2 + Erg_3 + Erg_4 eingegeben, dann versuchen wir wieder nach OpenCalais (Erg_10), Google Ajax Search API (Erg_11), Webwiki (Ergebnis_12) und DBPedia (Ergebnis_13), zu analysieren. Das finale Ergebnis von OpenCalais & Google Ajax Search API & Webwiki wird durch die Mehrheitsentscheidung von Erg_1 , Erg_2 , Erg_3 , Erg_4 , Erg_5 , Erg_6 , Erg_7 , Erg_8, Erg_9, Erg_10, Erg_11, Erg_12 und Erg_13 bestimmt:

Institute: University of Leipzig, City: Leipzig, Region: Saxony, Country: Germany, Longitude: -122.253425, Latitude: 37.869977.

OpenCalais + Google Ajax Search API + Google Search auf Wikipedia + DBPedia (OC + GA + WW + DBP):

Es wird einfach die Mehrheitsentscheidung nach Erg_1, Erg_2, Erg_3 und Erg_4 gerechnet. Und das finale Ergebnis sieht wie folgt aus:

Institute: Universität Leipzig, City: Leipzig, Region: Saxony, Country: Germany, Longitude: 12.39261, Latitude: 51.3283.

Nachdem alles erfolgreich analysiert wurd, kann man einfach die entsprechende Quelle-ID auswählen und dazu mit der Tabelle da_result, die alle richtige Information gespeichert hat, vergleichen bzw. mit der Metrik von der Kennungszahl von Precision, Recall und F Measure auswerten order wieder durch das Google Marker-Cluster / Google Chart darstellen.

3.2 Auswertung & Darstellung im Web-Framework

Dieser Abschnitt gliedert sich in zwei Teile.

Der erste Teil befasst sich mit der Auswertung der acht Analysemethoden, die nach der Metrik von Precision, Recall und F-Measure gerechnet werden und dann im Tabellen oder Chartlinie-Format analysiert werden. Eine weite Metrik ist diese, dass der Anteil der vier Fälle gerechnet wird, es werden Ort gefunden aber keine Institute, Institute aber keine Ort, beide Informationen werden nicht gefunden und der erfolgreiche Fall(Beide Information erfolgreich gefunden). Das zweite Teil umfasst die Darstellung im Web-Framework, wir zeigen die Papers in den Google Maps, sodass die gefundene Ort und Institute miteinander verbunden sind.

In unserem Beispiel werden 3643 Papierinformationen, deren Venue ‚sigmod' lautet, eingegeben, enthalten hier 1005 unterschiedliche. Weil man hier mehr Zeit benötigt, um auf den Webserver zu warten und viele Analysemethoden ausführen muss, ist das Geschehen sehr zeitraubend. Folgende Tabelle wird für die Auswertung der Metrik als Precision, Recall und F-Measure angezeigt:

Nur durch OpenCalais

	Precision	Recall	F-Measure
Institute:	72.58%	71.16%	71.86%
City:	49.01%	33.92%	40.09%
Country:	58.34%	41.35%	48.4%
Koordinaten(20km):	29.13%	18.08%	22.31%
Inst&City:	68.5%	25.51%	37.18%
Inst&Country:	74.33%	29.72%	42.46%
City&Country:	39.34%	18.57%	25.23%
Inst&City&Country:	56.9%	12.9%	21.03%

Nur durch Google Ajax Search API

	Precision	Recall	F-Measure
Institute:	57.83%	56.7%	57.26%
City:	53.14%	47.12%	49.95%
Country:	73.57%	71.85%	72.7%
Koordinaten(20km):	69.48%	61.88%	65.46%

Inst&City:	57.12%	36.07%	44.22%
Inst&Country:	72.22%	49.56%	58.78%
City&Country:	71.96%	43.89%	54.52%
Inst&City&Country:	68.74%	34.6%	46.03%

Nur durch Google Search auf Wikipedia (Web-Wiki)

	Precision	Recall	F-Measure
Institute:	36.69%	35.97%	36.33%
City:	36.23%	34.6%	35.4%
Country:	41.4%	39.78%	40.57%
Inst&City:	34.22%	27.57%	30.54%
Inst&Country:	37.14%	31.48%	34.08%
City&Country:	37.37%	31.96%	34.45%
Inst&City&Country:	33.04%	25.42%	28.73%

Nur durch DBPedia

	Precision	Recall	F-Measure
Institute:	41.87%	41.06%	41.46%
City:	46.02%	44.09%	45.03%
Country:	72.67%	70.97%	71.81%
Inst&City:	40.42%	24.73%	30.69%
Inst&Country:	63.62%	32.65%	43.15%
City&Country:	64.27%	40.27%	49.52%
Inst&City&Country:	58.04%	22.58%	32.51%

OpenCalais & Goolge Ajax Search API

	Precision	Recall	F-Measure
Institute:	76.67%	75.17%	75.91%
City:	61.85%	55.62%	58.57%
Country:	86.02%	82.99%	84.48%
Koordinaten(20km):	56.24%	54.15%	55.18%
Inst&City:	80.31%	45.85%	58.37%
Inst&Country:	90.91%	67.45%	77.44%
City&Country:	85.63%	53.57%	65.91%
Inst&City&Country:	88.67%	44.38%	59.15%

OpenCalais&GoogleAjax& Google Search auf Wikipedia

	Precision	Recall	F-Measure
Institute:	75.37%	73.9%	74.63%
City:	62.65%	57.87%	60.17%
Country:	87.08%	84.36%	85.7%
Koordinaten(20km):	53.5%	51.61%	52.54%
Inst&City:	80%	46.92%	59.15%
Inst&Country:	91.34%	67.06%	77.34%

City&Country:	86.84%	56.11%	68.17%
Inst&City&Country:	89.89%	46.04%	60.89%

OpenCalais&GA&WW&DBPedia

	Precision	Recall	F-Measure
Institute:	75.77%	74.29%	75.02%
City:	64.23%	61.78%	62.98%
Country:	88.59%	86.51%	87.54%
Koordinaten(20km):	51.52%	49.85%	50.67%
Inst&City:	81.56%	50.15%	62.11%
Inst&Country:	92.35%	68.43%	78.61%
City&Country:	87.02%	59.63%	70.77%
Inst&City&Country:	91.77%	49.07%	63.95%

OpenCalais+GA+WW+DBPedia

	Precision	Recall	F-Measure
Institute:	80.06%	78.49%	79.27%
City:	67.92%	65.2%	66.53%
Country:	90.59%	88.47%	89.52%
Koordinaten(20km):	67.69%	64.52%	66.07%
Inst&City:	86.11%	55.13%	67.22%
Inst&Country:	94.9%	72.83%	82.41%
City&Country:	89.53%	63.54%	74.33%
Inst&City&Country:	94.19%	53.86%	68.53%

Abbildung 3.5 Bewertung nach P.R.F.

Wie man aus obiger Tabelle ersieht, sind die Koordinaten von OpenCalais ungenau und auch in Bezug auf City und Country schaut es nicht so gut aus. Das Ergebnis nach den Instituten hingegen ist schon in Ordnung. Die Kombinationen von Institut und City, Institut und Country usw. sind auch unbefriedigend im Ergebnis. Für die Methode , Nur durch Google Ajax Search API' gilt, dass außer Institute alles andere Gewonnene besser als OpenCalais ist. Die Information zum Country ist schon mit Precision mit 73.57% und Recall mit 71.85% und F-Measure mit 72.7% als sehr gut zu bezeichnen. Die anderen Teile muss man alle noch verbessern. Bei der Methode , Nur durch Web-Wiki , sind alle Ergebnisse so unzureichend. Das Ergebnis von DBPedia ist schlechter als die Methode ‚Google Ajax Search API', aber die Precision, Recall und F-Measure zum Country erreichen immerhin als 70%, wie auch bei der Google Ajax Search API. Leider reichen andere Teile von F-Measure mit 50% noch

nicht aus. Bei den Methoden ‚OpenCalais & Google Ajax Search API' sieht das Ergebnis besser aus. Die F-Measure von Institute beträgt 75,91% und von Country ist sie mit 84,48% sehr hoch, auch die Kombination von Institute und Country mit 77,44% ist hoch. Aber für City und die anderen drei Kombinationen ist sie noch schlecht. Im einigen Fällen ist die Precision sehr hoch und Recall sehr niedrig, das bedeutet, es gibt viele Affiliationen, für die man keine Information bekommt. Die Methode ‚OpenCalais & Google Ajax & Web-Wiki' sieht besser aus, die F-Measure von allen Informationen außer den Koordinaten ist größer als 60%, besonders für Country ist sie mit 85,7% am höchsten, wohingegen die Institute mit 74.63% weniger gut abschneiden als mit OpenCalais & Google Ajax Search API. Die Kombinationen von Institute&Country liegen ganz nah bei 80% und sind damit auch sehr hoch. Aber man kann hier noch verbessern, man kann noch mit der Methode , DBPedia' ergänzen. Bei der Analyse aller vier Methoden zusammen mit analy_methode=7 sind die meisten Ergebnisse in Ordnung und einige Attributwerte sehr hoch. Aber es gibt noch einige Attribute, die sehen nicht so gut aus. Die gilt z.B. für die City, deren F-Measure nur 62,98% und die Kombination mit Inst&City auf 62.11% kommt, obwohl die Precision sehr hoch ist, aber das ist wohl in Ordnung. Leider ist die F-Measure von Institut geringer als mit Methode ‚OpenCalais&Google Ajax Search API' ausgefallen. Die bedeutet, dass die Methode von DBPedia und Wiki mehr falsche Informationen erhalten hat. Es bietet sich noch die analy_methode=8 an, die nur die Mehrheitsentscheidung von vier Ergebnissen der einzelnen Analysemethode wiedergibt. Überraschenderweise ist diese Auswertung die beste. Die F-Measure von Country steht ganz nah bei 90% und die meisten Werte sind größer als 70%.

Diese Auswertungszahlen können einfach die Parameter von Google Chart Playground [14] ersetzen und ausführen:

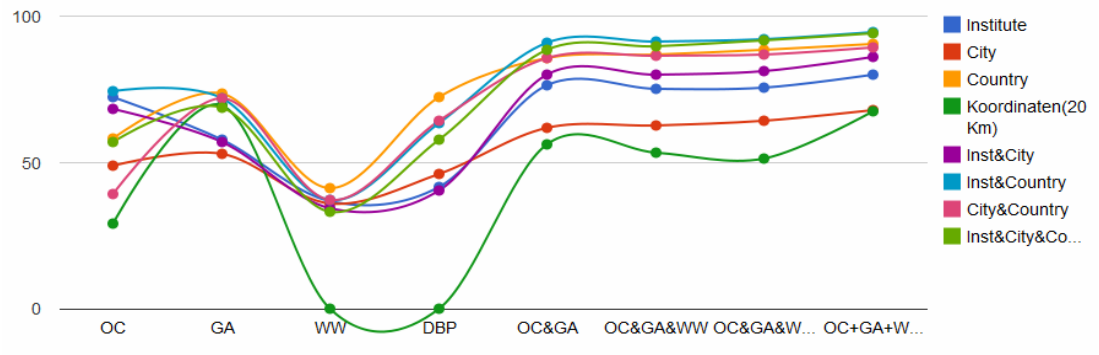


Abbildung 3.6 Precision-Darstellung nach verschiedenen Methoden

Wie im Abbildung 3.6 gezeigt, sind auf der Horizontalachse die Analysemethoden aufgetragen. Es wird ganz deutlich, dass für Web-Wiki die Ergebnisse schlechter sind als für andere Methoden und dass Google Ajax Search API mächtigere Analysewerkzeug ist. Die meisten Linien liegen durchschnittlich über 60% und die Kombinationsmethoden durchschnittlich nahe bei 80% .

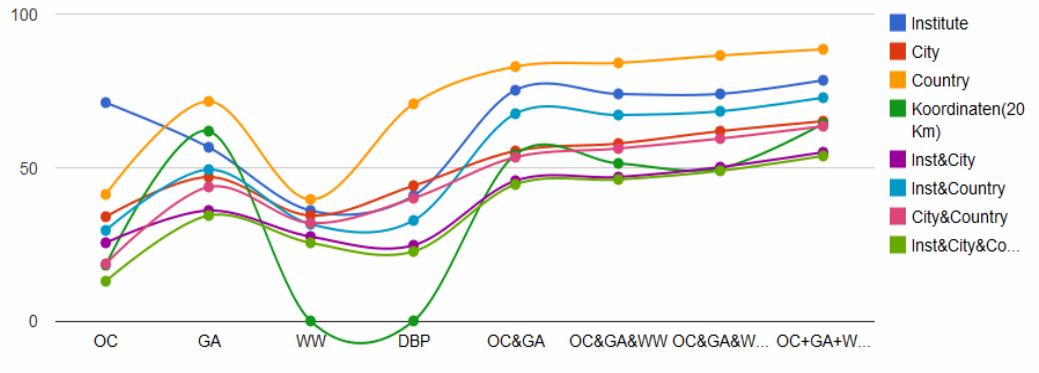


Abbildung 3.7 Recall-Darstellung nach verschiedenen Methoden

Wie in Abbildung 3.7 gezeigt wird, sind die Recall ganz streuend und schlechter als die Precision, d.h. es gibt noch einige Teile, die keine Informationen bekommen haben. Es kann einfach ersehen, dass durch die Kombinationsmethoden durchschnittlich ca. 60% erzielt werden.

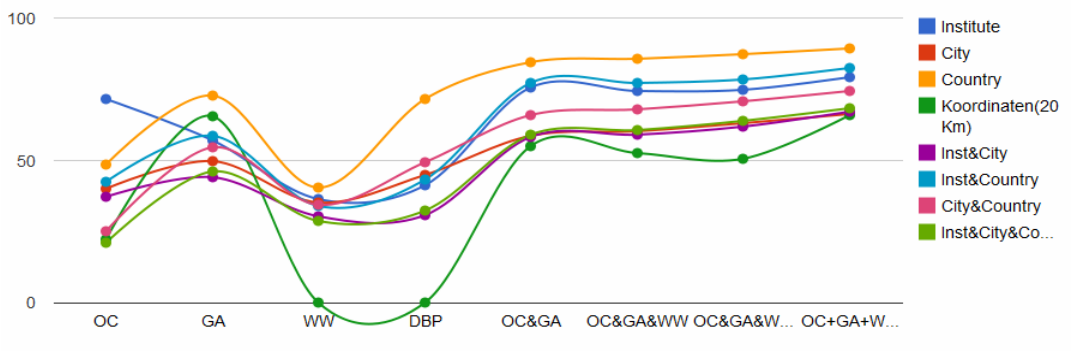


Abbildung 3.8 F-Measure nach verschiedenen Methoden

Die Abbildung 3.8 zeigt die F-Measure nach verschiedenen Analysemethoden. Mit den Kombinationsmethoden ist die F-Measure von Institut und Country sehr hoch und die anderen Werte liegen um die 70%. Die gelbe Linie bzw. die Linie über die Landesinformation liegt am höchsten, d.h. die Methoden sind sehr mächtig für die Erkennung von Country. Durch das Web-Wiki erhält man hingegen eine F-Measure von Country von weniger als 50%.

Die zweite Metrik ist so gestaltet, die Anteile nach vier Fällen zu rechnen:

Nur durch OpenCalais

Weder Ort noch Institut:	5.28%
Institute gefunden, aber keine Orte:	24.93%
Orte gefunden, aber keine Institute:	10.65%
Erfolgreich:	59.14%

Nur durch Google Ajax

Weder Ort noch Institut:	9.58%
Erfolgreich:	90.42%

Nur durch DBPedia

Weder Ort noch Institut:	2.05%
Institute gefunden, aber keine Orte!:	3.03%
Erfolgreich:	94.92%

Nur durch Web-Wiki

Weder Ort noch Institut:	49.17%
Institute gefunden, aber keine Orte:	7.14%
Orte gefunden, aber keine Institute:	0.39%
Erfolgreich:	43.3%

OpenCalais&GoogleAjax

Weder Ort noch Institut:	0.88%
Institute gefunden, aber keine Orte:	1.08%
Orte gefunden, aber keine Institute:	1.66%
Erfolgreich:	96.38%

OpenCalais&GoogleAjax&webWiki

Weder Ort noch Institut:	0.59%
Institute gefunden, aber keine Orte:	1.08%
Orte gefunden, aber keine Institute:	1.37%
Erfolgreich:	96.97%

OpenCalais&GoogleAjax&webWiki&DBPedia

Weder Ort noch Institut:	0.49%
Institute gefunden, aber keine Orte:	0.39%
Erfolgreich:	99.12%

OpenCalais+GoogleAjax+webWiki+DBPedia

Weder Ort noch Institut:	0.49%
Institute gefunden, aber keine Orte:	0.49%
Erfolgreich:	99.02%

Abbildung 3.9 Anteile nach vier Fällen

In einer Analysemethode ist die Google Ajax am besten. Das Ergebnis von Google Ajax Search API ist die Affiliation, dass entweder beide Informationen gefunden oder beide nicht gefunden werden. Aber in der Methode von OpenCalais & Google Ajax wird in einigen Teilen bei ‚Institute gefunden, aber keine Orte' ein Prozentsatz von 1,99% ausgewiesen. Dies bedeutet, dass einige Affiliationen durch Google Ajax Search API keine Information bekommen, hingegen bekommt man durch OpenCalais Institute, aber keine Orte. Die Abbildung 3.10 zeigt den Linien-Chart des Anteils der vier Fälle. Es gibt einen großen Sprung bei der Methode Web-Wiki. Der Fall ‚Erfolgreich' beträgt sofort weniger als 50%, aber der Fall ‚Beide Orte und Institute nicht gefunden' steigt schnell.

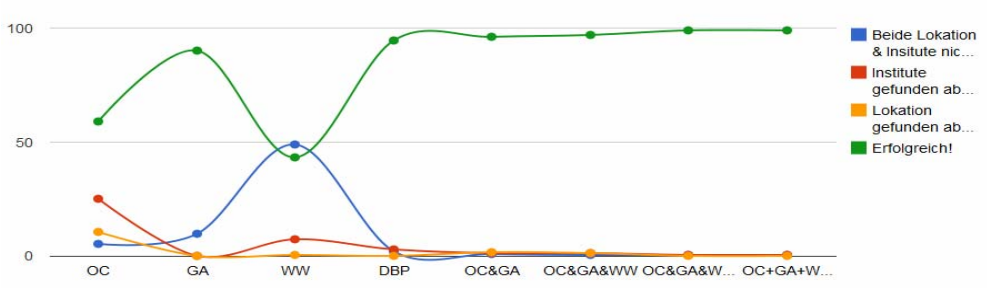
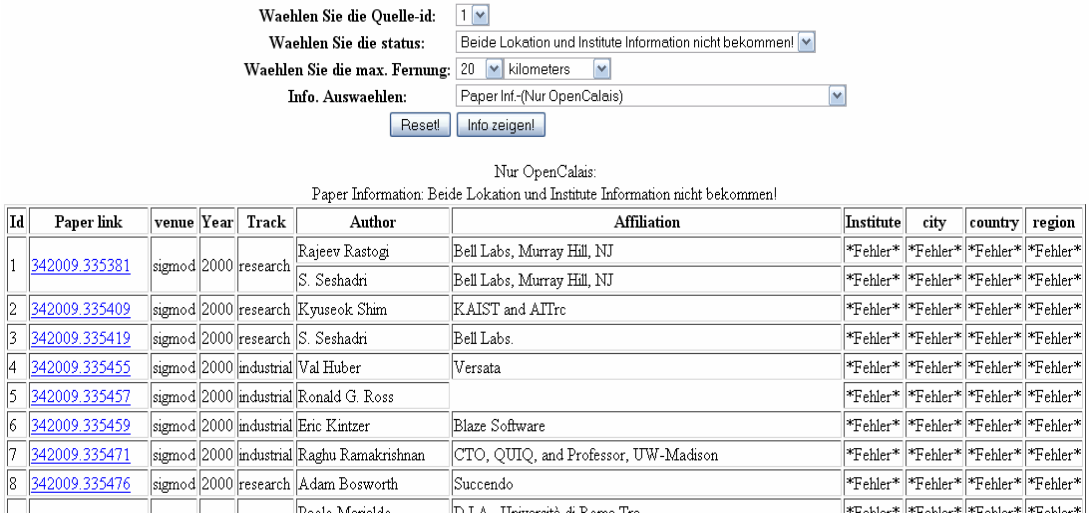


Abbildung 3. 10 Anteile der vier Fällen im Linie-Chart

Die entsprechenden Papierlisten in den vier Fällen werden im Menü von ‚Wählen Sie den Status' und ‚Info. Auswählen' zusammen bestimmt und dann wird das Botton von ‚Info zeigen' angeklickt. Zum Beispiel wählen wir den Status als ‚Beide Orte und Institute nicht gefunden!' und Inf. Auswählen als ‚Paper Inf. –(Nur OpenCalais)' :



Id	Paper link	venue	Year	Track	Author	Affiliation	Institute	city	country	region
1	342009.335381	sigmod	2000	research	Rajeev Rastogi	Bell Labs, Murray Hill, NJ	*Fehler*	*Fehler*	*Fehler*	*Fehler*
					S. Seshadri	Bell Labs, Murray Hill, NJ	*Fehler*	*Fehler*	*Fehler*	*Fehler*
2	342009.335409	sigmod	2000	research	Kyuseok Shim	KAIST and AITrc	*Fehler*	*Fehler*	*Fehler*	*Fehler*
3	342009.335419	sigmod	2000	research	S. Seshadri	Bell Labs.	*Fehler*	*Fehler*	*Fehler*	*Fehler*
4	342009.335455	sigmod	2000	industrial	Val Huber	Versata	*Fehler*	*Fehler*	*Fehler*	*Fehler*
5	342009.335457	sigmod	2000	industrial	Ronald G. Ross		*Fehler*	*Fehler*	*Fehler*	*Fehler*
6	342009.335459	sigmod	2000	industrial	Eric Kintzer	Blaze Software	*Fehler*	*Fehler*	*Fehler*	*Fehler*
7	342009.335471	sigmod	2000	industrial	Raghu Ramakrishnan	CTO, QUIQ, and Professor, UW-Madison	*Fehler*	*Fehler*	*Fehler*	*Fehler*
8	342009.335476	sigmod	2000	research	Adam Bosworth	Succendo	*Fehler*	*Fehler*	*Fehler*	*Fehler*

Abbildung 3.11 Paperlisten (Nur OpenCalais, Beide Info nicht bekommen)

Wie in Abbildung 3.11 dargestellt, werden alle Paper in einer Tabelle aufgelistet und jedes Paperlink mit der Webseite vom ACM verlinkt. Wir können die Paper noch weiter in Googlemaps nach eingegebenden Suchbedingungen mit der MarkerCluster Technik darstellen. Angenommen, wir wählen die Quelle=1 und als Bedingung die Paper, die im Jahr 2000 erstellt sind. Die Analysemethode ist die ‚Nur OpenCalais'. dann wird per Default ca. die Hälfte der gesamt gefundene Affiliationen in Google Maps gezeigt. Bei der Auswahl des gewählten Papers wird seine entsprechende Position in Google Maps mit entsprechenden gefundenem Foto darstellt.

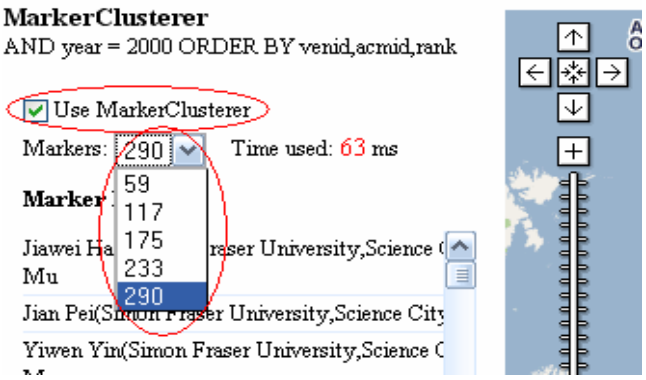


Abbildung 3.12 MarkerCluster in Google-Maps

Es gibt noch eine weitere Anzeigetechnik, die Google-Chart, die auch von Google angeboten wird. In dieser Arbeit wird die Anzahl von Papers nach Land und Jahr präzise angezeigt. Alle statistischen Daten werden im Google-Spreadsheet gespeichert, dessen Gestaltung und Funktion der von Office-Excel entspricht, das Online arbeitet und in dem Sheets mit verschiedene Rechten geführt werden können. In der oberen rechten Ecke können verschiedene Diagramme ausgewählt werden, links das Scattplott-Diagramm, in der Mitte das Histogramm und rechts die parallelen Koordinaten. Man kann ein bestimmtes Land am Graphen auswählen oder entfernen. Man kann das Playbutton im unteren Teil anklicken oder die Balken dazwischen über das Jahr auswählen.

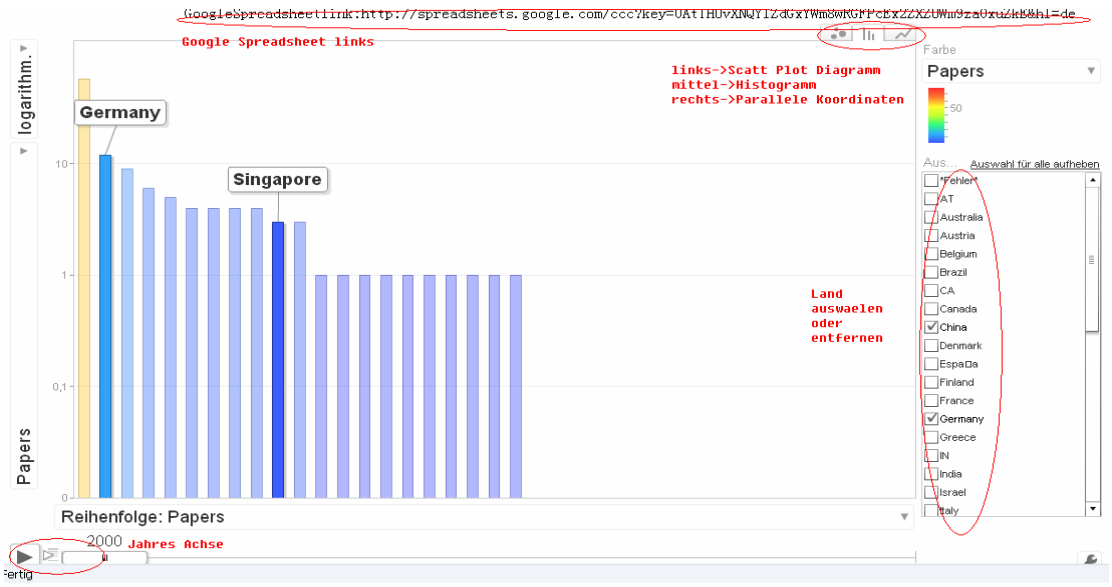


Abbildung 3.13 Google Chart(Histogramm)

IV. Implementierung

4.1 System-Beschreibung

Es wird mit PHP programmiert und als Datenbank MySQL genutzt.

Folgende sind die wichtigen Bedingungen, um das Programm erfolgreich auszuführen:

1. Internet
2. XAMP PHP Package initialisieren. Das Package enthält PHP, Apache und Mysql.
3. Richtige Informationen über Institute und Orte zur entsprechenden Affiliation vorbereiten, die in die Tabelle ‚da_result' importiert werden.
4. DBPedia-Standarddaten von der Infobox über Universität, Country, City und Region, die später im page_clean gespeichert wird.
5. Die API von OpenCalais, Google AJAX Search API beantragen, und aktualisieren der Datei ./Modell/config.php. Die Webseite der Beantragung von OpenCalais API: http://www.opencalais.com/user/register, die Webseite der Beantragung von Google Ajax Search API: http://code.google.com/intl/de-DE/apis/ajaxsearch/signup.html, und nicht zu vergessen die Kontrolle,, ob die Logininformationen von Datenbank bzw. Benutzername, Password und Datenbanksname im config.php gültig sind.

Systeme Initialisierung:

1. Die richtige Information in die ‚da_result' manuell importieren.
2. Die DBPedia-Standarddaten in die Tabelle manuell importieren und bearbeiten, d.h. die Tabelle page_university und page_clean vollständigen.

4.2 ER-Diagramm & Datenbankstrukturen

Unser Ziel ist es, wir geben eine Affiliation ein, dann bestimmen wir mit verschiedene Analysemethoden die Informationen von Institute, Country, City, Region und manchmal zusätzlich die Koordinaten oder das Bild, am Ende wird das Ergebnis in der Datenbank gespeichert.

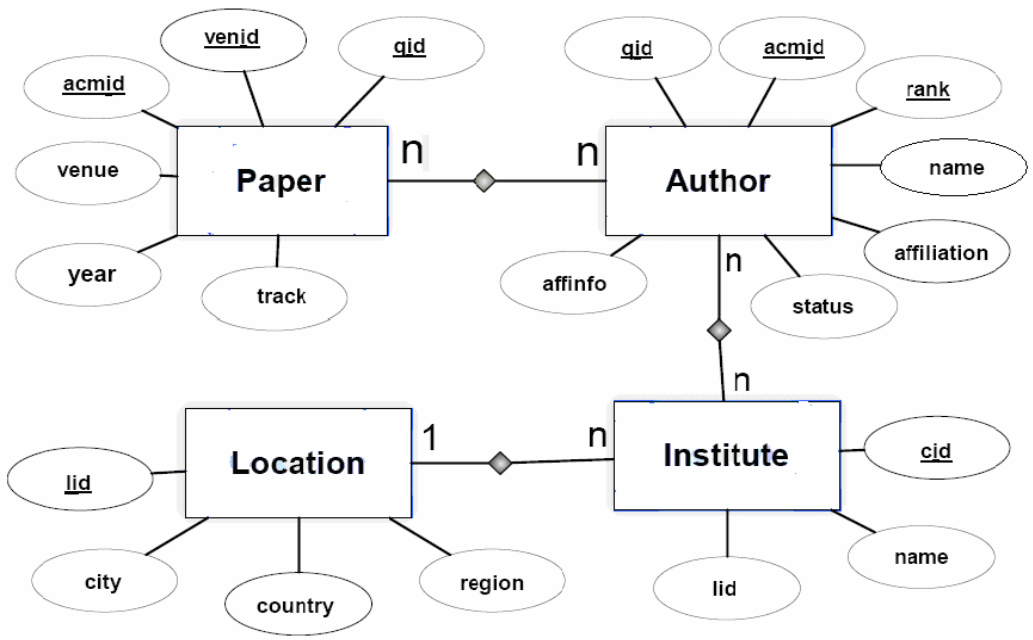


Abbildung 4.1 ER-Diagramm

Wie im obigen ER-Diagramm gezeigt, jedes Paper kann mehr als einen Autor haben, so kann auch jeder Autor mehr als ein Paper erstellen. D.h. die Beziehung zwischen Paper und Autor ist ‚n zu n'. Nehmen wir an, dass es für jeden Autor nur eine Affiliation gibt. (Weil im allgemeinen Fall ein Autor mehrere Affiliationen haben kann). Weil wir viele Analysemethoden ausführen, bekommen wir eine Vielzahl von Ergebnissen, es verweist jede Affiliation zur unterschiedlichen Instituten und Orten, deswegen ist die Beziehung zwischen Autor und Instituten ‚n zu n'. Jedes Institut gehört zu nur einem Ort und mehr Autoren können in einem gleichen Institut arbeiten. An einem Ort können sich mehrere Institute befinden. So generieren sich die Datenbankstrukturen als folgende:

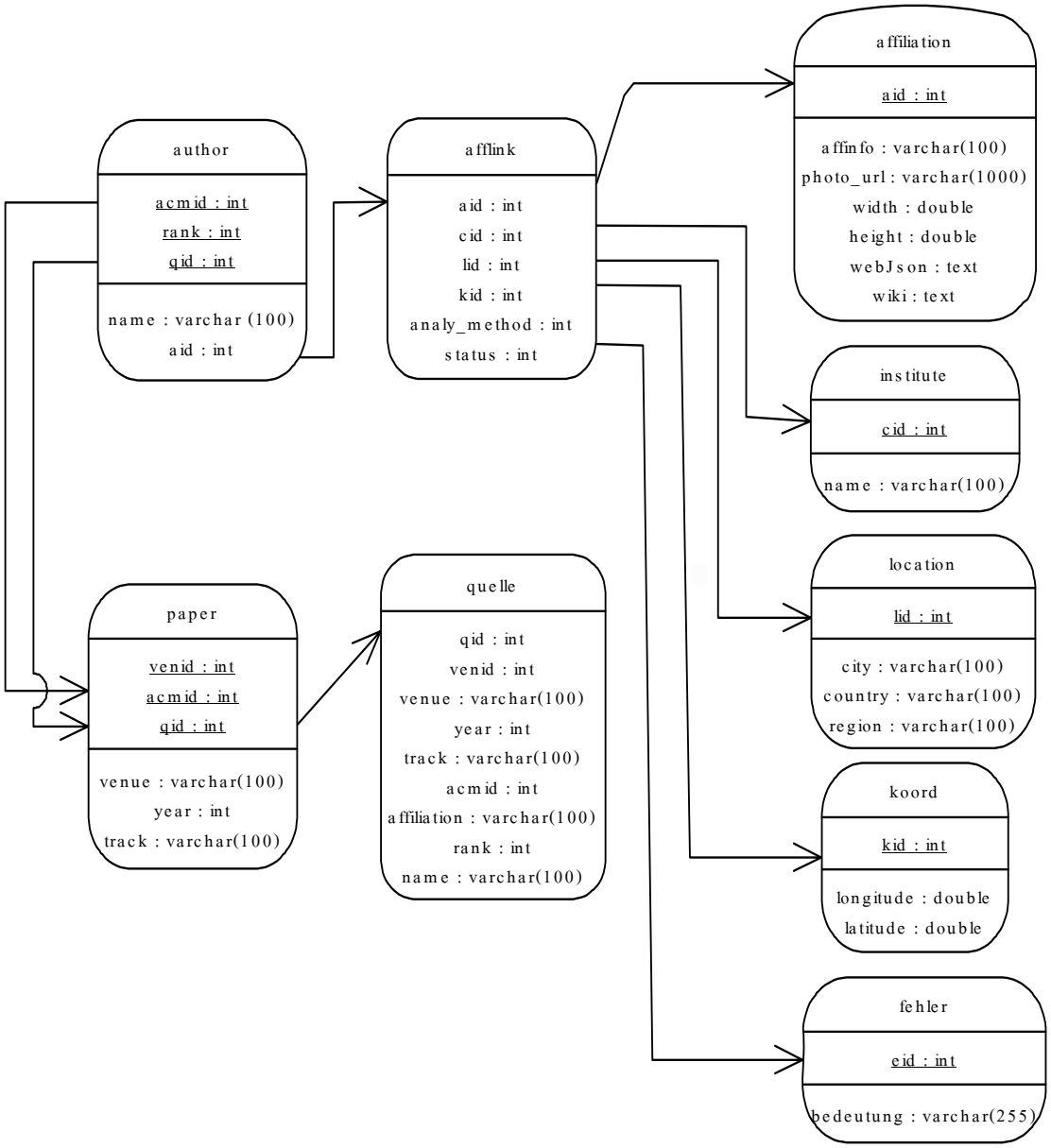


Abbildung 4.2 Tabellenbeziehungen der Affiliationanalysespeicherung

Alle Pfeiler sind die Foreignkey von Tabellen, die Attribute ‚analy_method' von der Tabelle ‚afflink' sind die entsprechenden Variablen der Analysemethode. Diese Tabellenstrukturen sind nicht perfekt, weil zwischen dem Paper und dem Autor ‚n zu n ‚-Beziehung besteht, die wir vorher schon analysiert haben. D.h., wenn ein Autor zur mehreren verschiedenen Papers gehört oder zwei Mal der gleiche Autor eingegeben wird, wird die Tabelle Autor mehrere Duplikate enthalten. Die Beziehung zur Tabelle ‚da_result', die alle richtigen Informationen gespeichert, ist ganz einfach. Man braucht nur die entsprechende Autor- und Paper-Information zur Affiliation zu verbinden:

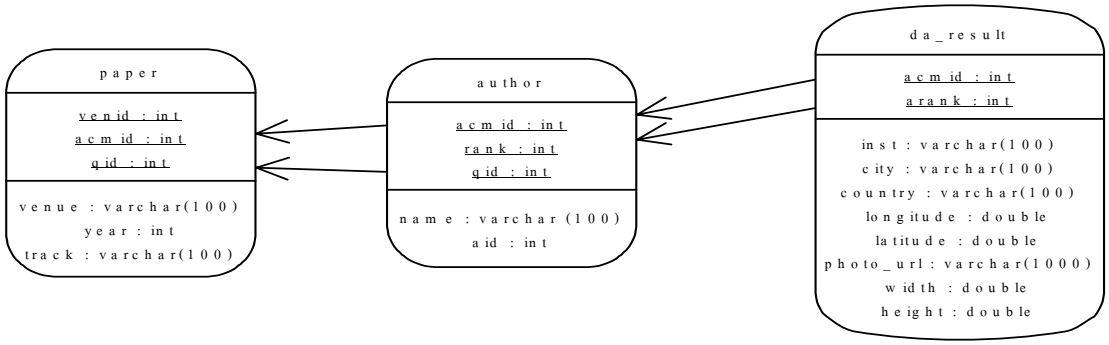


Abbildung 4.3 Tabellenbeziehung von ‚da_result'

Es gibt noch drei benötige Standardtabellen von DBPedia:

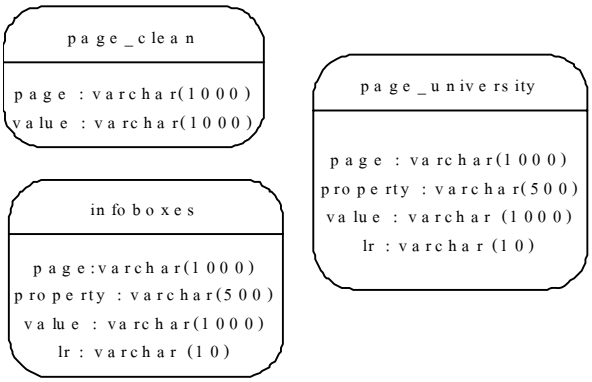


Abbildung 4.4 Standardtabellen der DBPedia

Die originalen Informationen von Infoboxes kann man unter der Hompage von DBPedia herunterladen, nach der einzigen Verfeinerungsphase und Informationsfilterung werden sie in der Tabelle page_clean gespeichert.

Z.B. die originale Datensätze von Infoboxen:

page : Doctor%27s_Ambulance

property: wikiPageUsesTemplate

value : Template:infobox_ambulance_company

lr : r

Durch Verfeinerung werden die Daten in page_clean so gespeichert:

page : Doctor's Ambulance

value : Template:infobox_ambulance_company

Dass heißt ‚%27' und '_' werden zum Leerzeichen ersetzt. Nachdem wir die Eingabe erhalten haben, analysieren wir im ersten Schritt über die ‚page_university' Tabelle für die Institute:

SELECT DISTINCT(page) FROM page_university WHERE MATCH (page) AGAINST (“….”) OR MATCH(value) AGAINST (“….”)

Die Parameter von zwei Klammern sind die originale Affiliation. Diese SQL Anfrage nutzt die MATCH AGAINST Technik und speichert alle gefundenen Institute, falls schon eine passende Institute erhalten wurde. Ansonsten versuchen wir wieder mit der Tabelle ‚page_clean’, zu analysieren, leider es gibt sehr viele unterschiedliche ‚Property’ über Institute:

SELECT * FROM page_clean WHERE MATCH (page) AGAINST ("…..") AND (value LIKE "%company%" OR value LIKE "%organization%" OR value LIKE "%school%" OR value LIKE "%education%" OR value LIKE "%university%" OR value LIKE "%college%" OR value LIKE "%conference%" OR value LIKE "%department%").

Die Property von ‘Company’, ’Organisation’ usw., alles gehört zu den Instituten. Die Anfragen nach Ort bzw. City, Country und Region sind ganz analog.

Im Folgenden werden wir einige relevante SQL-Anfragen von True-Positive oder False-Positive der entsprechenden Attribute erklären (Quelle-ID als 1 und max. Entfernung bis 20 km):

1).Die Rechnungsweise von Institute-True Positive

Am Anfang führen wir SQL aus, um die Affiliation mit den entsprechenden Instituten von der Datenbank auszulesen. Die Berechnung von True Positive von Institute ist kompliziert. Im ersten Schritt ersetzen wir die Substrings mit , of ’ und ’-’ zum Leerzeichen, weil es keine Bedeutung hat. Dann trennen wir das Wort durch Leerzeichen und wählen immer alle ersten Buchstaben und wandeln die kleinen Buchstaben als Abkürzungsstring um. Z.B. ist der Abkürzungsstring von ‚University Leipzig’ und ‚University of Leipzig’ ‚ul’. Dann prüfen wir, ob die Abkürzungsstring Teilstring voneinander sind. Wenn Ja, werden wir diese zwei Strings Äquivalenz zuweisen, sonst werden wir sie als unterschiedlich beurteilen. Die beiden Strings von ‚University of Leipzig’ und ‚Informatik Universität Leipzig’ sind deshalb äqivalent, weil der erste Abkürzungsstring ‚ul’ ein Teilstring von dem zweiten Abkürzungsstring

‚iul' ist. Diese Rechenweise ist mächtig, aber theoretisch nicht korrekt, bei zwei verschiedenen Worten mit gleichem ersten Buchstaben zum Beispiel wird es Fehler geben. Aber man wird bemerken, im Paktischen werden die meisten richtigen Institute mit verschiedenen Schreibweisen mit dem richtigen Ergebnis beurteilt, d.h., die obige Fall tritt kaum ein.

2).Die Rechnungsweise von City-True Positive

Die City True Positive sind viel einfacher als die Institute-True Positive, es wird nur der originale String von City verglichen, ob er Teilstring von einem anderen ist, weil die meisten Citys in der Bezeichnung ganz kurz sind und man keinen Abkürzungsstring umzuwandeln braucht.

3).Die Rechnungsweise von Country-True Positive

Diese ist ganz analog City-True Positve. Der Unterschied liegt nur in der Ersetzung der Attribute ‚City' durch ‚Country' und dem Prüfen des Strings des Landes, ob er ein Teilstring von einem anderen ist.

4).Die Rechnungsweise von Koordinaten-True & False Positive

Am Anfang lesen wir die entsprechende Koordinaten von der Tabelle aus. Dann rechnen wir den Abstand der zwei gefundenen Positionen und vergleichen, ob er größer oder kleiner als 20km ist. (Vorher haben wir die max. Entfernung auf 20km gesetzt). Wenn der Zwischenabstand größer als 20km ist, dann gehört das Ergebnis zu False Positive, sonst gehört es zu True Positive. Die Formel zu zwei gegebenen Koordinatenabständen lautet:

Parameter: $lat1, $lon1, $lat2,$lon2

```
$theta = $lon1 - $lon2;
$dist = sin(deg2rad($lat1)) * sin(deg2rad($lat2)) + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) * cos(deg2rad($theta));
$dist = rad2deg(acos($dist));
```

$miles = $dist * 60 * 1.1515;

(deg2rad, rad2deg sind die Funktionen, die zwischen Grad und Radius umzuwandeln)

Fall die zurückgelieferte Einheit als Kilometer(km) angefordert wird:

Liefert das Ergebnis als $miles * 1,609344 aus.

Fall die zurückgelieferte Einheit als miles angefordert wird:

Liefert das Ergebnis als $miles * 0,8684 aus.

Fall die zurückgelieferte Einheit als nautical angefordert wird:

Liefert das Ergebnis direkt $miles aus.

5).Die Rechnungsweise von Institute-False Positive

Die False Positive von Institute ist analog der True Positive Institute. Im ersten Schritt wird wieder eine SQL-Anfrage ausgeführt, um die Affiliation mit den entsprechenden Instituten von der Datenbank auszulesen. Die Abkürzungsbehandlung ist die wie vorher. Wenn der originale String nicht ein Teilstring von einem anderen ist und auch die Abkürzung kein Teilstring von einem anderen ist, dann gehören diese beide Institute zum False Positive von Institute.

6).Die Rechnungsweise von City-False Positive

Dieses Verfahren ist ganz analog wie City-True Positive, es prüft, ob die beiden Citys nicht Teilstring voneinander sind und die richtige Information von City nicht leer ist. Die andere Rechnungsweise von True-Positive ist ganz analog der von City-True/False Positive oder Institute-True/False Positive, das Unterschied liegt nur im WHERE Prädikanten.

Mit der zweiten Auswertungsmetrik wird der Anteil der vier Fälle gerechnet. Das entsprechende SQL lautet wie folgt:

SELECT count(*) as affs, COUNT(distinct a.aid) as affstrings,link.status,f.bedeutung, 100*COUNT(distinct a.aid)/(SELECT count(distinct aid) FROM author a2 WHERE qid=1) as perc FROM author a,fehler f,afflink link WHERE a.aid=link.aid AND link.status=f.eid AND a.qid=1 AND link.analy_method=$analy_method GROUP BY

link.status,f.bedeutung';

Die Attribute ‚affs' bedeuten die Anzahl der Affiliationen, ‚affstrings' bedeutet die Anzahl der verschiedene Affiliationen, ‚status' bedeutet den Status der vier Fällen, z.B. beide Informationen nicht erhalten, nur Institute erhalten, nur Orte erhalten und Erfolgreich! (beide Informationen erhalten), ‚f.bedeutung' ist die konkrete Information des Status, ‚perc' ist der Anteil der Affiliationen zu den entsprechende Fällen.

4.3 Problem & Zukunftsarbeit

Im diesem Abschnitt werden wir uns mit anhängigen Problemen, die in dieser Arbeit noch nicht vollständig abgeklärt werden konnten und solchen, die möglicherweise noch im Code enthalten sind, befassen. Problemlösungsansätze nennen, die aus unserer Sicht in der Zukunft zu einer Weiterentwicklung der Thematik beitragen könnten..

1). Das Teilstringtesten von Abkürzungsstrings ist nicht vollständig korrekt.

In unserem Algorithmus ist die True Positive von Institute nicht vollständig richtig. Wir haben nur die Abkürzungen dahingehend beurteilt, ob es Teilstrings voneinander gibt.

(QID:1)True Positve von Institute!

index	Affiliation	result_inst	analy_inst
1	School of Computing Science, Simon Fraser University	Simon Fraser University	Simon Fraser University
2	Hewlett-Packard Laboratories, Palo Alto, California	HP	Hewlett-Packard Laboratories
3	School of Computer Science, Carnegie Mellon University, Pittsburgh, PA	CMU	Carnegie Mellon University Pittsburgh
4	Lucent Bell Labs, 600 Mountain Avenue, Murray Hill, NJ, Computer Science and Engg., Tndian Institiue	Bell	Lucent Bell Labs
5	Database Systems Lab, SERC, Indian Institue of science, Bangalore 560012,INDIA, Lucent Bell Labs, 60	Bell	Lucent Bell Labs Database Systems Lab
6	Microsoft Research, One MicrosoftWay, Redmond, WA	Microsoft	Microsoft

Abbildung 4.5 Beispiele von verschiedenen Instituten mit gleicher Bedeutung

Wie die Abbildung 4.5 zeigt, ist das Abkürzungsstringstesten sehr mächtig. Z.B. sind die zweite und dritte Affiliation zwischen result_inst und analy_inst ganz unterschiedlich, aber zwischen den Abkürzungsstrings existiert ein Teilstring. (Der Teilstring mit ‚of' wird einfach weggelassen, weil er keine Bedeutung ergibt.)

Index 2, Abkürzung_result_inst: h (oder hp), Abkürzung_analy_inst='hpl'

Index 3, Abkürzung_result_inst: c(oder cmu) , Abkürzung_analy_inst= ‚cmup'

Deswegen werden sie als gleiche Institute beurteilt und zum True Positive gezählt. Aber es gibt auch Problematisches bei einer einzigen Affiliation. So zum Beispiel bei den zwei Instituten ‚IBM' und ‚InFomix Software', obwohl die beiden Informationen totale unterschiedliche sind, gelangt man bei dem Algorithmus zu gleicher Bedeutung. Die Abkürzungsstring von result_inst lautet: ‚i' und die analy_inst lautet ‚is', so ist ‚i' ein Teilstring von ‚is', aber die Vorteile überwiegen die Nachteile.

2). Nicht vollständige Korrektur aus der da_result.

Weil die manuell vorgegebene richtige Information in der ‚da_result' zur entsprechenden Affiliation nicht vollständig richtig ist, beeinflusst sie die Qualität von Precision, Recall und F-Measure.

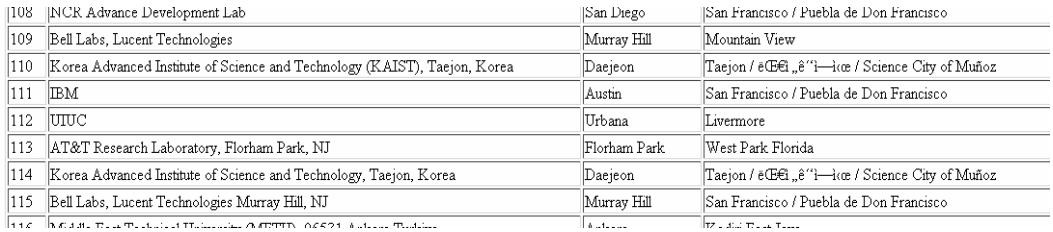
index	Affiliation	result_country	analy_country
288	East China Normal University, Shanghai, China	USA	China
296	Bell Laboratories, Alcatel-Lucent, Lisle, IL, USA	India	USA

Abbildung 4.6 Falsche Information vom Land in der Tabelle ‚da_result'

Wie in Abbildung 4.6 dargestellt, kann man einfach sehen, dass die Landinformation in der da_result Tabelle nicht ganz korrekt ist, aber man hat erfolgreich die richtige Information von der analy_country bekommen. Mit dieser Technik kann man einfach die Information von da_result oder die originale Affiliation korrigieren.

3). Anzeigeprobleme in HTML, die nicht alle Sprache gleichzeitig unterstützt.

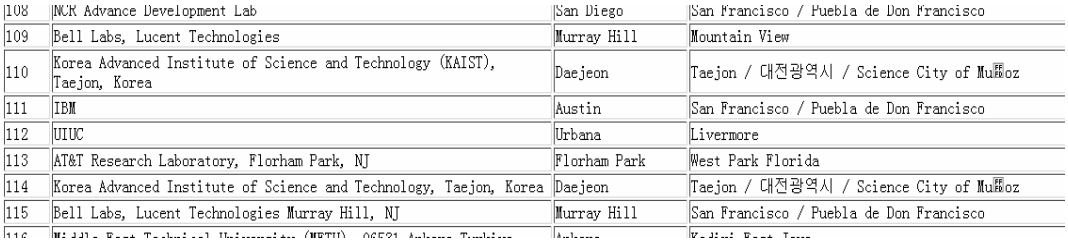
Zum Beispiel bekommt man nach die Analysemethode der OpenCalais& Google Ajax Search API &Web-Wiki folgende Institute:



108	NCR Advance Development Lab	San Diego	San Francisco / Puebla de Don Francisco
109	Bell Labs, Lucent Technologies	Murray Hill	Mountain View
110	Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea	Daejeon	Taejon / ëŒ€ì „ê´‘ì—ì‹œ / Science City of Muñoz
111	IBM	Austin	San Francisco / Puebla de Don Francisco
112	UIUC	Urbana	Livermore
113	AT&T Research Laboratory, Florham Park, NJ	Florham Park	West Park Florida
114	Korea Advanced Institute of Science and Technology, Taejon, Korea	Daejeon	Taejon / ëŒ€ì „ê´‘ì—ì‹œ / Science City of Muñoz
115	Bell Labs, Lucent Technologies Murray Hill, NJ	Murray Hill	San Francisco / Puebla de Don Francisco

Abbildung 4.7 Fehler bei Kodierungsproblemen

Wie in der Abbildung 4.7 gezeigt, wird die 5. oder 8. Affiliation als falsche angezeigt. Bei obigen Seiten wird als Charset die ‚ISO-8859-1' eingesetzt. Wenn man den Parameter mit ‚unicode' ausgewählt hat, wird erfolgreich gezeigt.



108	NCR Advance Development Lab	San Diego	San Francisco / Puebla de Don Francisco
109	Bell Labs, Lucent Technologies	Murray Hill	Mountain View
110	Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea	Daejeon	Taejon / 대전광역시 / Science City of Muñoz
111	IBM	Austin	San Francisco / Puebla de Don Francisco
112	UIUC	Urbana	Livermore
113	AT&T Research Laboratory, Florham Park, NJ	Florham Park	West Park Florida
114	Korea Advanced Institute of Science and Technology, Taejon, Korea	Daejeon	Taejon / 대전광역시 / Science City of Muñoz
115	Bell Labs, Lucent Technologies Murray Hill, NJ	Murray Hill	San Francisco / Puebla de Don Francisco

Abbildung 4.8 Richtige Anzeige von anderer Sprache

So wird in der Abbildung 4.8 die analy_city richtig angezeigt. Es kann erfolgreich koreanisch angezeigt werden. Mit der Kodierung von ‚utf-8' können die meisten europäischer Buchstaben nicht kodiert werden. Zum Beispiel werden mit ‚ä, ü, ß' falsche kodiert, gezeigt wie auch im Abbildung 4.8, ersichtlich die 114te Affiliation. Bei den meisten Anfragen an ‚Google Ajax Lokation Search API' wird die entsprechende Information in der gefundenen Landsprache zurückgeliefert. Bei Web-Wiki hingegen wird immer der englische Text ausgegeben, weil nur die englische Seite von Wiki aufgesucht wird.

4) Es werden nicht alle Sprache unterstützt.

Dieses Problem ist ein Teilproblem der Kodierungsprobleme. OpenCalais, Google Ajax Search API, web-Wiki und DBPeida unterstützen die meisten asiatischen Sprachen noch nicht. Wenn sie unterstützt werden, ist wieder die Berechnung von Precision, Recall und F-Measure problematisch. Bei dem Vergleich mit zwei Worten,

die von gleicher Bedeutung sind und sich in verschiedenen sprachlichen Strings präsentieren, ergeben sich Probleme. Ich denke, dass sich die Unterstützung der mehr sprachigen Textanalyse von OpenCalais, Google Ajax Search API usw. in Zukunft weiterentwickelt, beim Vergleich der Strings kann dann eine weitere Übersetzungskomponente entwickelt bzw. eine Ziel sprache definiert werden. Zum Beispiel kann man dann alle Texte durch diesem Komponente ins Englische übersetzen. Dann wird der englische String analysiert oder auswertet. Bei der Anzeige des Ergebnisses von Institute und Country kann man sich einfach vorher in der Übersetzungs- oder Analysephase für die entsprechende Sprache entscheiden und dann die entsprechenden Charsets der HTML anpassen. Das ganze Verfahren wird in Abbildung 4.13 (auf der nächsten Seite) angezeigt.

5) Synonyme Information noch nicht perfekt erkannt.

Die Cityangaben von ‚München', ‚Munich', und ‚Muenchen' sind tatsächlich von gleiche Bedeutung. D.h., es gibt sehr viele Schreibweise aus gleichen oder verschiedenen Sprachen. Ein weiteres typisches Beispiel, man kann schwer zwischen dem chinesischen München und dem deutschen München mit einem String vergleichen. Hier habe ich in der config.php einige synonyme Informationen im Array $syn_country, $syn_city, $syn_inst gespeichert, am Anfang der Qualitätsrechnung wird die entsprechende Information der Tabelle erneuert, aber wie bei dem Kodierungsproblem ist es so, dass man viele synonyme Datensätze schwerlich manuell im Array bearbeiten kann, deswegen ist diese Technik nicht perfekt.

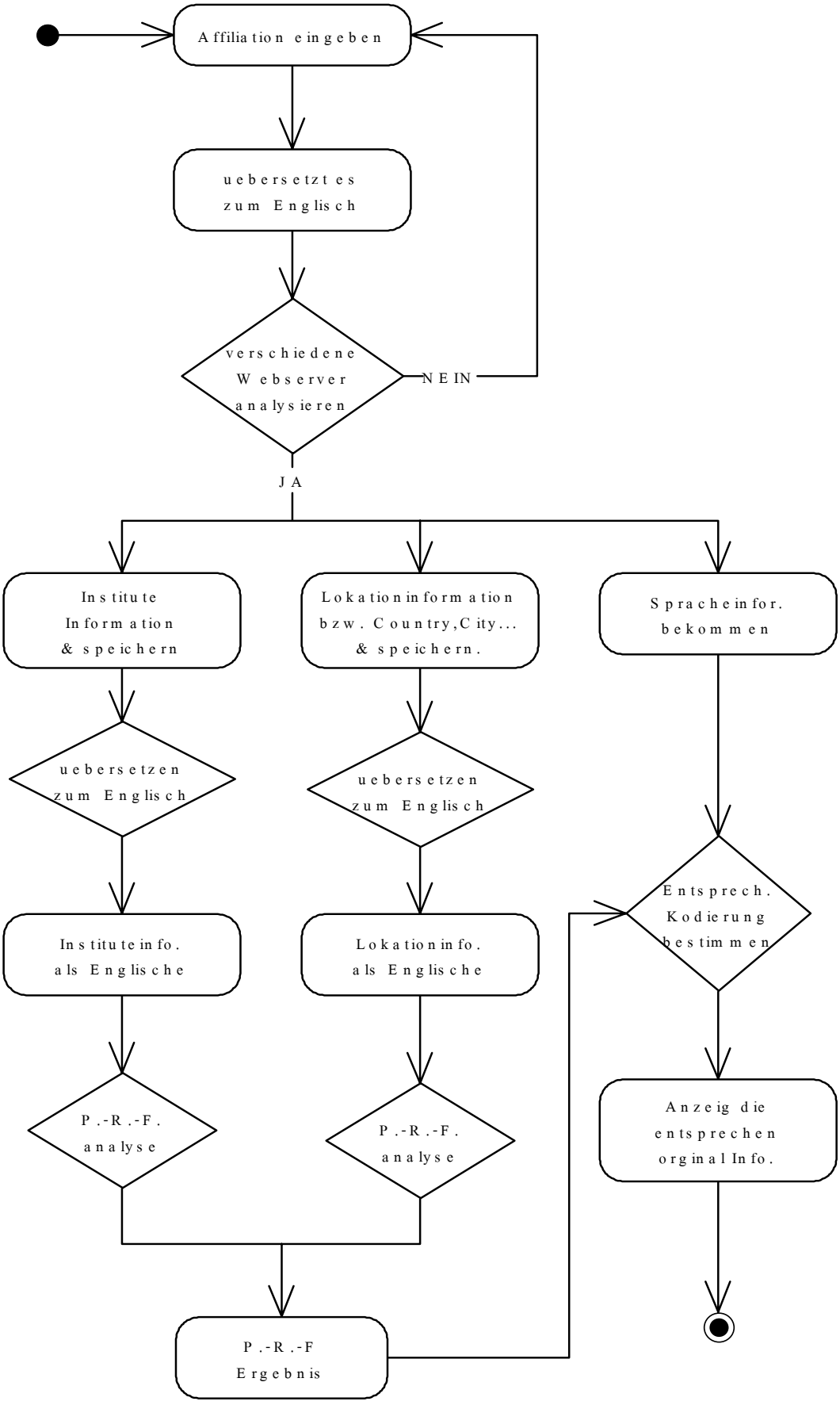


Abbildung 4.9 Datenfluss für Zukunftsprogramm, Zielsprache in Englisch

V. Zusammenfassung

Die vorliegende Bachelorarbeit präsentiert Web-basierte Ansätze, um beliebige Affiliation-Angaben aus wissenschaflichen Papieren mit verschiedenen Analysemethoden nach Instituten und Orten zu untersuchen und auszuwerten. Ziel ist es, unterschiedliche Affiliation-Angaben, die dasselbe Institut bezeichnen, zusammenzuführen. Mit diesem Ergebnis kann dann weiter gearbeitet werden, z.B. in bibliografischen Analysen und geografischen Darstellungen.

Eine Menge von Affiliations der Autoren wissenschaftlicher Papiere werden einer Analyse unterzogen. Es werden vier unterschiedliche Analysemethoden in Betracht gezogen: OpenCalais, Google Search, Google Search auf Wikipedia, sowie DBPedia. Ausgenommen DBPedia, das lokal in ein DBMS geladen wird, arbeiten die Methoden web-basiert via Web-APIs.

In der Bachelorarbeit werden Darstellungstechniken vorgestellt, um das Ergebnis der Analyse zu visualisieren. Mittels MarkerClusterer können die Affiliations der untersuchten Papers in Google Maps dargestellt werden. Man kann einfach ersehen, welches Land die meisten Papers erstellt. Mittels Google Charts können die Anzahl von Papers von Institutionen, Orten und Ländern auf unterschiedlichste Weise verglichen werden. Durch diese Graphen können auch die Trends der Papers in den vergangenen Jahren veranschaulicht werden.

Außer der Darstellung spielt die Auswertung der gefundenen Ergebnissen eine große Rolle. Hier wird mit Precision, Recall und F-Measure die Korrektheit der Web-basierten Affiliation-Untersuchungsmethoden bewertet. Hierzu benötigt man zusätzlich manuell erstellte korrekte Informationen über Institute und Orte, die mit den gefundenen Ergebnissen verglichen werden. Eine Kombination auf Basis von Mehrheitsentscheiden über alle vier Analysemethoden erzielte dabei das beste Ergebnis.

Abbildungsverzeichnis

Literaturverzeichnis

[1] OpenCalais Beschreibung: http://www.opencalais.com/about

[2] OpenCalais Dokument Viewer: http://viewer.opencalais.com/

[3] OpenCalais Input Content: http://www.opencalais.com/documentation/calais-web-service-api/forming-api-calls/input-content

[4] Homepage von ACM Webseite : http://portal.acm.org/citation.cfm

[5] Homepage Google AJAX Search API : http://code.google.com/intl/de-DE/apis/ajaxsearch/

[6] Homepage DBPedia: http://dbpedia.org/About

[7] Thomas Schandl, Semantischer Content mit OpenCalais, Semantic Web Company Wien, 2009

[8] Aumüller, D. & Rahm, E. Web-based Affiliation Matching 14th International Conference on Information Quality (ICIQ 09), Potsdam, Germany, 2009

[9] Prof. Dr. Uwe Quasthoff, Vorlesung Folien Information Retrieval 7-8 SS 2009 Institut für Informatik Universität Leipzig

[10] ImageChart und interactives Chart: http://code.google.com/intl/de-DE/apis/charttools/

[11] Google Geo Developers Blog

http://googlegeodevelopers.blogspot.com/2009/04/markerclusterer-solution-to-too-many.html

[12] Google Chart Playground http://code.google.com/apis/ajax/playground/?type=visualization#line_chart

[13] Pereira, D.; Ribeiro-Neto, B.; Ziviani, N. & Laender, A. Using web information for creating publication venue authority files Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries, 2008, 295-304

[14] J. C. French, A. L. Powell, and E. Schulman. Using clustering strategies for creating authority files. Journal of the American Society for Information Science, 51(8):774–786, 2000.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

---	---	---
Ort	Datum	Unterschrift