Christian Biemann

Extraktion von semantischen Relationen aus natürlichsprachlichem Text mit Hilfe von maschinellem Lernen

Zusammenfassung: Inhalt der vorliegenden Arbeit ist die Entwicklung eines Lernverfahrens, das aus großen Textkorpora semantische Relationen automatisch extrahiert.

Den Kern des Verfahrens bildet die Iteration von Suchschritt und Verifikationsschritt, in denen in gesuchter Relation stehende Wörter gefunden und überprüft werden. Auf diese Weise ist es möglich, mit wenigen bekannten Wörtern eine große Anzahl in derselben Relation stehende Wörter zu gewinnen. So können mit wenig Aufwand große Listen von Wörtern erstellt werden, die in einem semantischen Zusammenhang stehen.

Nach der Skizzierung des Algorithmus werden theoretische Vorhersagen bezüglich der für das Verfahren geeigneten Relationen getroffen, sowie der Ablauf modelliert.

Einige mit einer Implementierung des Verfahrens erzielte Ergebnisse werden für die Relation der Personennamen vorgestellt, evaluiert und diskutiert, des Weiteren werden Ausblicke und Verbesserungsmöglichkeiten angegeben.

1 Semantische Relationen

1.1 Arten von semantischen Relationen

Für das automatische Verstehen der Bedeutung eines Textes ist es unerlässlich, sämtliche kompositionalen Bedeutungsbeziehungen, also semantischen Relationen, als solche zu erkennen. Wie viele semantische Relationen existieren, ist nicht bekannt, da die Granularität fast beliebig hoch angesetzt werden kann, um alle Facetten der Bedeutungskomposition zu erfassen. Eine technische Umsetzung muss daher immer einen Kompromiss zwischen ausreichender Repräsentation und Machbarkeit schließen.

Es wird zwischen paradigmatischen und syntagmatischen Beziehungen unterschieden. Paradigmatische Beziehungen haben mit mentalen Assoziationen zu tun und beziehen sich auf das Verhältnis der sprachlichen Elemente im gesamten Sprachsystem, während die syntagmatischen Beziehungen die Aneinanderreihung der sprachlichen Elemente zur linearen, beobachtbaren Sprache bestimmen.

Viele der aus den linguistischen Semantik bekannten Relationen lassen sich nur sehr schwer automatisch auffinden, auf der anderen Seite existieren für die automatisch auffindbaren Relationen teilweise keine Namen in der Semantik. Die linguisti-

sche Taxonomie kann jedoch als Leitlinie und Maß dafür gelten, wie viel an dieser Stelle schon erreicht wurde beziehungsweise noch erreicht werden muss.

In dieser Arbeit werden lediglich semantische Relationen behandelt, die sich durch Pattern Matching auf der Wortfolge oder der Abfolge aufgrund flacher Eigenschaften zugewiesener Tags erkennen lassen.

1.2 Durch Pattern Matching auffindbare semantische Relationen

Ein Tag ist ein Kürzel, das einem Symbol (hier im Fall der natürlichen Sprache: einem Wort) aufgrund dessen Eigenschaften zugewiesen wird. Das Tagging erfolgt durch einen Tagger, der die zu taggende Symbolfolge (einen Text) einliest und den Symbolen (hier: Wörter, Satzzeichen, Numerale etc.) passende Tags aus dem Tagset (Gesamtheit aller Tags) zuweist.

Ein Pattern ist ein regulärer Ausdruck bestehend aus Symbolen oder Tags. Tritt in der Symbolfolge das Muster des Patterns auf, so spricht man von einem Match.

Die Pattern werden dazu verwendet, semantische Relationen im Text zu erkennen. Matcht ein Pattern, so liegt vermutlich die mit diesem Pattern assoziierte Relation vor.

Die klassischen semantischen Relationen wie Synonymie, Antonymie, Hyponymie usw. sind paradigmatischer Natur und können kaum von Pattern erkannt werden. Sätze wie "Sommer ist das Gegenteil von Winter." sind auch in großen Korpora kaum zu finden. Syntagmatische Relationen hingegen eignen sich für Pattern, z.B. seien die Head-Modifier-Relation, die Relation, die Vor- und Nachnamen zu Personennamen verknüpft, oder auch die Mörder-Opfer-Relation ("Oswald tötete Kennedy", siehe (Duclaye 2002)) als wenige Beispiele genannt.

Bei der Konstruktion der Pattern muss beachtet werden, dass stark generalisierende Pattern zu oft matchen und somit nur sehr unscharf die gewünschte Relation liefern, zu spezielle Pattern hingegen matchen zu selten und liefern nur unvollständige Daten.

Im Folgenden soll ein Verfahren vorgestellt werden, das mit Hilfe von unscharfen Pattern und einem Verifikationsschritt Elemente semantischer Relationen mit hoher Genauigkeit liefert.

2 Der Pendel-Algorithmus

Das nachfolgend beschriebene Verfahren stützt sich auf folgende Beobachtung: Vornamen und Nachnamen stehen in natürlichsprachlichem Text oftmals hintereinander. Es sollte daher möglich sein, von bekannten Vornamen ausgehend zugehörige Nachnamen zu finden und umgekehrt. Der naive Ansatz, einfach die im Text rechts von

Vornamen stehenden Wörter als Nachnamen und die links von Vornamen stehenden Wörter als Vornamen zu bezeichnen, schlägt fehl, auch wenn man die Großschreibung dieser Wörter verlangt.

Ein Verifikationsschritt ist nötig, der die durch Nachbarschaftsregeln gefundenen Namenskandidaten überprüft und mit hoher Genauigkeit Namen von Nichtnamen unterscheidet: Belegstellen des Namenskandidaten werden daraufhin überprüft, ob der Vornamenskandidat (bzw. Nachnamenskandidat) genügend oft vor (bzw. hinter) einem schon bekannten Nachnamen (bzw. Vornamen) im Text auftritt. Ist dies der Fall, so wird der Name gelernt und kann zum Auffinden von weiteren Namen verwendet werden.

2.1 Voraussetzungen an die Relation

Es bleibt zu klären, wie die für das Verfahren geeigneten Relationen beschaffen sind, bevor dieses im Detail beschrieben wird.

In diesem Zusammenhang interessieren n-stellige, mindestens zweistellige Relationen $R \subseteq A_1 \times A_2 \times ... \times A_n$ mit $A_i \cap A_j = \emptyset$ für $i \neq j$.

Der Algorithmus soll folgendes leisten:

- Auffinden der Relation, gegeben einige Elemente aus $A_1,...,A_n$
- Auffinden möglichst vieler einzelner Elemente der Mengen $A_1,...,A_n$.
- Auffinden möglichst vieler Tupel $(a_1,...,a_n)$, die in der Relation R stehen.

Im natürlichsprachlichen Fall sind Elemente der Mengen A_i Wörter, Instanzen der Relation R sind Ausdrücke, die die Relation R realisieren.

Im Folgenden werden die Relationen sowie die Mengen A_i in Großbuchstaben mit selbsterklärenden Namen bezeichnet.

DEFINITION: BIPARTITER GRAF

Ein Graf (V,E), V Knotenmenge, E Kantenmenge heißt bipartit, wenn es Partitionen $V_1 \subseteq V$, $V_2 \subseteq V$, $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$ gibt, so dass für alle Elemente $e=(v_1,v_2) \in E$ gilt: $v_1 \in V_1$, $v_2 \in V_2$.

DEFINITION: N-PARTITER SCHICHTGRAF

Ein Graf G (V,E), V Knotenmenge, E Kantenmenge heißt n-partiter Schichtgraf, wenn es Partitionen $V_1, V_2, ... V_n$, $\cup V_i = V$, $V_i \cap V_j = \emptyset$ für $i \neq j$, gibt, so dass für alle Paare von Partitionen V_i und V_{i+1} (i=1,..,n-1) gilt: Die Einschränkung von G auf $V_i \cup V_{i+1}$ ist ein bipartiter Graf und für alle Paare von Partitionen V_i, V_j, $|i-j|>1$ gilt: es existieren keine Kanten der Form (v_i, v_j) mit $v_i \in V_i$, $v_j \in V_j$.

Die Elemente der Mengen $A_1,..,A_n$ können als Knoten, das benachbarte Auftreten als Kanten eines Grafen aufgefasst werden. Da die Mengen $A_1,..,A_n$ paarweise disjunkt sind, ist die Struktur des Grafen die eines n-partiten Schichtgrafen (im Fall der zweistelligen Relation: bipartit).

Instanzen der Relation bedeuten im Grafen Knoten eines Weges von A_1 nach A_n, die mit einer Kante verbunden sind.

Nur Relationen im oben ausgeführten Sinne sind geeignet für das Pendel-Verfahren, darüber hinaus muss der n-partite Schichtgraf, der die Relation beschreibt, eine gewisse Dichte vorweisen, dazu später mehr.

2.2 Algorithmus

Die Abbildung gibt den Algorithmus auf halbformale Weise an:

```
Initialisiere Wissen, Regeln, neue Items
Solange es neue Items gibt:
  Für alle neuen Items i
      Hole Text mit i
      Finde Kandidaten in Text aufgrund Wissen und Regeln
      Verifiziere Kandidat k:
            Hole Text mit k
            Bewerte k aufgrund Text
      neue Items sind Kandidaten ab Schwellwert/mit höchsten Bewertungen
      Erweitere Wissen durch neue Items
```

Abb. 1: Algorithmus.

Im Weiteren werden die Elemente $a_i \in A_i$, die aus Sicht des Algorithmus in der Relation $R=(A_1,...,A_n)$ stehen, mit Items bezeichnet.

Der Algorithmus startet mit einem gewissen Grundwissen, dazu gehören einige Elemente der Mengen A_i (Beispielitems) und Regeln, durch die aus der Umgebung von bekannten Symbolen (Wörtern) neue Symbole gelernt werden können. Des Weiteren gehört zum Grundwissen eine Liste von häufigen Wörtern, die zur Klassifikation beitragen können, wie zum Beispiel Pronomen und Artikel. Nach der Initialisierung beginnt eine zweifach verschachtelte Programmschleife. In der äußeren Schleife wird jedes neue Item i behandelt. Für das Item i werden Belegstellen aus dem Korpus gesucht, auf welche die Regeln angewendet werden. Auf diese Weise entsteht eine Kandidatenliste von möglichen neuen Items, mit Klassifizierungen auf Zugehörigkeit zu A_1, $A_{2,}$... oder A_n.

In der inneren Schleife werden die Kandidaten überprüft: Es werden Belegstellen für die Kandidaten gesucht, wenn der Kandidat k hier aufgrund derselben Regeln oft genug genauso klassifiziert wird wie zuvor, wird k zum Wissen und zu den neuen Items des nächsten Durchgangs der äußeren Schleife hinzugefügt.

Auf diese Weise liefern die jeweils in einem Schritt gelernten Items die Kandidaten für den nächsten Schritt. Der Algorithmus terminiert, wenn keine neuen Items mehr gefunden werden.

2.2.1 Grundwissen, Tagset und Regeln

Das Grundwissen teilt sich in einen problemunabhängigen und einen problemabhängigen Teil. Problemunabhängig sind häufige Füllwörter wie Artikel, Pronomen und Präpositionen. Je mehr verschiedene Arten dieser Wörter unterschieden werden sollen, desto mehr Tags sind nötig.

Im problemabhängigen Teil des Grundwissens befinden sich die Beispiele für die Mengen, über die die gesuchte Relation definiert ist. Für jede der Mengen Ai, sowie für die verschiedenen Klassen des problemunabhängigen Grundwissens wird ein Tag benötigt, um die Pattern definieren zu können.

Des Weiteren werden Tags für flache Eigenschaften wie Großschreibung, Kleinschreibung, Numerale und Satzzeichen vergeben. Wörter können mehrere verschiedene Tags erhalten, beispielsweise wird ein großgeschriebenes Wort, das im Grundwissen enthalten ist, sowohl das Tag für Großschreibung als auch das Tag seiner bekannten Klasse zugewiesen bekommen. Die Pattern dienen dazu, Wörter, die einer bestimmten Bedingung genügen, und deren Umgebung mit dem Pattern matcht, auf Zugehörigkeit zu einer der vorher erwähnten Menge A_i, im Folgenden auch Klassen genannt, zu klassifizieren. Im Folgenden wird eine Regel A B* C $\rightarrow$ D so verstanden, dass bei einer vorliegenden Sequenz A B C dem Wort an der mit dem * markierten Stelle auch die Eigenschaft D zugewiesen werden kann.

2.2.2 Suchschritt

Im Suchschritt wird anhand von bekannten Items versucht, neue Items aufzufinden. Hierbei wird wie folgt vorgegangen: Zu einem bekannten Item werden Beispielsätze aus dem Korpus gesucht. So wird sichergestellt, dass in den Sätzen jeweils mindestens ein bekanntes Item vorkommt. Die Annahme ist dabei, dass neben diesem Item andere, noch unbekannte Items stehen, die mit dem bekannten Item in der gesuchten Relation stehen.

Nach der Annotation der Beispielsätze mit Tags werden die Regeln auf die Tags angewandt. Es werden Kandidaten für neue Items geliefert, zusammen mit den vermu-

teten Klassifizierungen, die im Verifikationsschritt überprüft werden. Dies erfolgt für alle im letzten Schritt neu gewonnenen Items.

2.2.3 Verifikationsschritt

Um zu ermöglichen, dass Regeln mit hoher Abdeckung aber geringer Schärfe verwendet werden können, ist ein Verifikationsschritt erforderlich. Zu jedem Kandidaten k werden Beispielsätze aus der Datenbank abgefragt, die wiederum annotiert werden. Zu beachten ist hierbei, dass dem jeweiligen k noch nicht das Tag seiner vermuteten Klasse zugewiesen wird. Im Gegensatz zum Suchschritt interessieren an dieser Stelle nur die Klassifikationen von k. Überschreitet das Verhältnis der Klassifikationen von k, die der im Suchschritt gefundenen Klasse von k entsprechen, zur Gesamtanzahl der Vorkommen von k einen gewissen Schwellwert, so wird k mit seiner Klassifikation ins Wissen aufgenommen und gilt fortan als Item. Im nächsten Suchschritt wird dann k zum Finden von neuen Kandidaten benutzt.

2.2.4 Einordnung in bekannte Verfahren

Das Verfahren implementiert Bootstrapping, dessen Eigenschaft es ist, aus wenig Anfangswissen unter Zuhilfenahme von neu erlerntem Wissen weiteres Wissen zu gewinnen. Vergleichbare Ansätze lassen sich in (Riloff 1999), (Collins 1999) und (Duclaye 2002) finden. (Riloff 99) beschreibt einen Zwei-Level-Bootstrapping Mechanismus, der in alternierenden Schritten Items und Extraktionsregeln lernt und mit Hilfe einer Meta-Ebene nur die besten Items in die nächste Iteration übernimmt.

Collins und Singer benutzen in (Collins 1999) einerseits die Buchstabenfolge der Items, andererseits deren lokalen syntaktischen Kontext, um zwei Klassifikatoren zu lernen, die sich gegenseitig alternierend trainieren.

2.3 Theoretisches Verhalten

Die gewünschten Eigenschaften eines Verfahrens zum Auffinden semantischer Relationen in einem Korpus sind hoher Recall, also wenn möglich das Auffinden aller Elemente der beteiligten Mengen bei hoher Precision, es sollen nur die gewünschten Elemente und keine unerwünschten gefunden werden.

2.3.1 Beschränktes Wachstum bei erfolgreichem Verlauf

Im Folgenden soll angenommen werden, dass das Verfahren mit hoher Precision die meisten auffindbaren Items extrahiert.

Als Vereinfachung sei angenommen, dass jedes Item durch Suche und Verifikation genau N neue Items liefern kann. Pro Iterationsschritt sollte sich die Anzahl neu klassifizierter Items jeweils um N vervielfachen, was exponentielles Wachstum bedeutet. Nun ist aber die Gesamtanzahl der Items beschränkt und je höher die schon erreichte Abdeckung, desto mehr der N Items werden schon bekannt sein und sind somit nicht mehr neu klassifizierbar: es tritt Sättigung ein.

Das mathematische Modell von beschränktem Wachstum exponentieller Natur ist das logistische Wachstum, und es kann erwartet werden, dass sich die Anzahl von Items pro Schritt sich dementsprechend verhält.

2.3.2 Fehlerfortpflanzung

Bei einem unüberwachten Lernverfahren wie diesem kann es vorkommen, dass Wörter fälschlicherweise als der Relation zugehörig klassifiziert werden. Da diese in den nächsten Schritten als Grundwissen für weitere Klassifikationen verwendet werden, können sich Fehler fortpflanzen, indem falsche Items erneut falsche Items liefern.

Angenommen, ein Startitem ist falsch. Im nächsten Schritt produziert dieses Item unkorrekte Items, die in den Folgeschritten ihrerseits weitere falsche Items liefern. Die Fehlerrate jedoch bleibt in diesem Szenario zunächst konstant, weil in der Wachstumsphase die richtig klassifizierten Items jeweils viele neue richtige Items liefern, und steigt nur an, wenn falsche Items aufgrund richtiger Items gelernt werden. In der Sättigungsphase kann ein Anstieg der Fehlerrate erwartet werden, weil die Sättigung nur bei richtigen Items eintritt, während das Sättigungsmanko für falsche Items nach wie vor groß ist.

2.3.3 Anforderungen an die Symbolfolge

Es wurden Vorversuche durchgeführt, um zu untersuchen, welchen Anforderungen die Symbolfolge genügen muss, um mit dem beschriebenen Algorithmus durch Pattern auffindbare Relationen zu extrahieren. Betrachtet wurde zunächst die Dichte des der Relation zugehörigen bipartiten Grafen, die diesem Zusammenhang die durchschnittliche Menge von Ausgangskanten eines jeden Knoten darstellt. Ein dünn besetzter Graf wird aufgrund des Verifikationsschrittes, der zur Bestätigung der Klassifizierung eines Knotens mehrere adjajeszente, schon bekannte Knoten braucht, kaum vollständig abgesucht werden. Auf einem dichten Graf hingegen sollte das Verfahren hohen Re-

call erreichen, da hier ein noch unklassifizierter Knoten der einen Partition mit hoher Wahrscheinlichkeit mit einigen bekannten Knoten der anderen Partition zusammenhängt.

Des Weiteren interessierte die Toleranz gegenüber der Unschärfe der Regeln, beziehungsweise, welchen Anteil von falsch zutreffenden Regeln das Verfahren verkraftet.

Die generierten Pseudotexte bestanden aus jeweils 100 unterscheidbaren Pseudowörtern aus den Mengen A, B und F, wobei die gesuchte Relation $R \subseteq A \times B$ war (Versuche zwischen 50 und 750 Elementen pro Klasse lieferten dieselben quantitativen Aussagen.). Nach Generierung des Grafen von R wurden die Itempaare aus $A \times B$ zufällig aneinandergereiht, dazwischen befanden sich jeweils fünf Wörter aus F. Abhängig von einer Fehlerrate wurden falsche Itempaare aus $A \times F$ bzw. $F \times B$ eingestreut. Als Regeln standen $A\ X^* \rightarrow B$ und $X^*\ B \rightarrow A$ zur Verfügung, als Grundwissen je 5% der Mengen A und B, begonnen wurde mit einem Startitem. Variiert wurden die durchschnittliche Anzahl der Kanten pro Knoten, sowie die Fehlerrate. Eine Fehlerrate von 50% bedeutet, dass die Regeln in der Hälfte der Fälle falsche Items liefern. Eine Übersicht über die Versuche gibt die Abbildung unten.

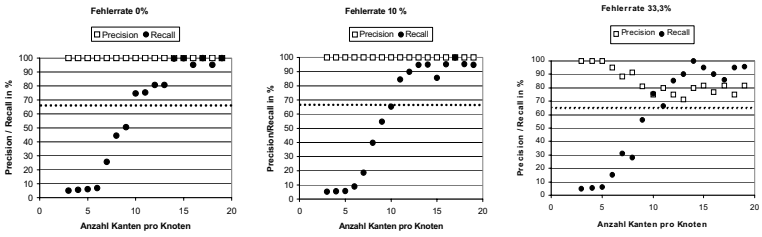

Abb. 2: Precision und Recall für Pseudotext für verschiedene Dichten und Fehlerraten.

Zu beobachten ist ein Phasenübergang für die Anzahl der Knoten pro Kanten, der sich von 7 bis 16 erstreckt[1]. Daraus wird deutlich, dass jeder Knoten durchschnittlich 17 Verbindungen in die andere Knotenpartition haben muss, damit nahezu alle Knoten gefunden werden, also nahe 100% Recall erreicht wird.

Aus der nächsten Abbildung ist die Toleranz des Verfahrens gegenüber Fehlern in den Daten beziehungsweise unscharfen Regeln ersichtlich. Die gepunktete Linie gibt die Baseline für die Precision an: bei 66,6% werden alle Elemente aus den Mengen A, B und F gefunden.

[1] Die Spanne des Phasenüberganges erwies sich auch bei weiteren Versuchen unabhängig von der Fehlerrate

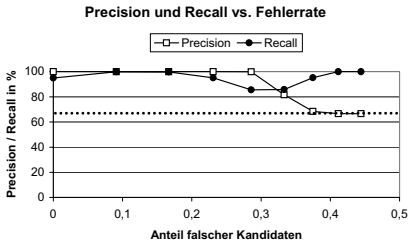

Abb. 3: Precision und Recall bei variierender Fehlerrate für 16 Kanten pro Knoten[2].

Die Ergebnisse bleiben bis zu Fehlerraten von 20% konstant, danach fällt der Recall ab, da manche Zielitems aus A und B aufgrund der Fehler nicht mehr oft genug mit der anderen Klasse verbunden sind. Ab 27% Fehlern verfällt die Precision und erreicht ab einer Fehlerrate von 0,36 nur noch die Baseline.

Zusammengefasst: Das Verfahren kann die Elemente zweier Mengen, einer durch Pattern erkennbaren Relation stehen extrahieren, wenn diese durchschnittlich 17 mal im Text instantiiert werden und wenn die zur Verfügung stehenden Regeln mit einer Genauigkeit von 80% zuschlagen.

Da nie auf dem gesamten Grafen, sondern nur auf lokalen Teilstrukturen operiert wird, sind die Ergebnisse für Grafen mit mehr Knoten derselben lokalen Dichte beziehungsweise für größere Korpora verallgemeinerbar.

2.4 Regeln lernen

In der natürlichen Sprache wird eine semantische Relation häufig nicht nur durch einen einzigen Pattern erkannt, sondern manifestiert sich in verschiedenen Abfolgen von Wörtern und Wortarten, denen jeweils ihr eigenes Regelmuster zugrunde liegt. Zum Beispiel kann die Relation PERSONENNAME in folgenden Varianten auftreten:

- PERSONENNAME_1$\subseteq$VORNAME×NACHNAME
- PERSONENNAME_2$\subseteq$VORNAME×VORNAME×NACHNAME
- PERSONENNAME_3$\subseteq$VORNAME×ADELSBEZ×NACHNAME
- ...

Der Pendel-Algorithmus kann nur Instanzen der Unterrelation erkennen, wenn er über eine Regel mit dem entsprechenden Pattern verfügt. Da es einerseits mühsam ist, die verschiedenen Regeln pro Relation per Hand aufzustellen, andererseits nicht klar ist, inwieweit diese Regelmenge die Instanzen der Relation im Korpus abdecken, bietet

[2] Die Messpunkte dieser und vorheriger Abbildung sind jeweils gemittelt über 20 Versuche

sich ein automatisches Verfahren zum Regellernen an. Voraussetzung dafür ist ein annotierter Trainingstext, in dem viele Instanzen der gesuchten Relation enthalten sind.

2.4.1 Konstruktionsschritt

Pro Klasse, die in der gesuchten Relation enthalten ist, wird je ein Konstruktionsdurchlauf durchgeführt. Im oben genannten Beispiel wären dies drei Schritte, einmal für die Klasse VORNAME, einmal für ADELSBEZ (mit den Elementen "von", "van", "di" etc.) und einmal für NACHNAME.

Nacheinander werden alle Wörter der jeweiligen Klasse im annotierten Text gesucht und es werden alle möglichen Pattern eines gewissen Längenintervalls um das bekannte Vorkommen herum konstruiert.

Für längere Texte ergibt das eine große Menge an potentiellen Regeln, wovon die meisten jedoch ungeeignet für den Pendel-Algorithmus sind, da zu allgemein, z.B. die Regel GR* GR KL → VN[3]. Einschränkungen auf die Patternkonstruktion sind möglich, beispielsweise kann gefordert werden, dass im Pattern mindestens ein Tag der Mengen der gesuchten Relation enthalten ist.

2.4.2 Verifikationsschritt

Die so gefundenen Pattern werden im Folgenden anhand des Trainingstextes verifiziert. Die Regeln werden gegen den Text gematcht und es wird gezählt, wie oft jede Regel richtige und falsche Aussagen über ihr jeweiliges Zielwort machte. Liegt das Verhältnis dieser beiden Werte über einem gewissen Schwellenwert, so wird die Regel als geeignet in den anschließenden Pendelprozess übernommen.

3 Experimente

In diesem Abschnitt werden zwei Experimente zu Personennamen, die mit einer Implementierung des Pendel-Algorithmus durchgeführt wurden, im Einzelnen vorgestellt. Weitere Experimente, u.a. zu Inseln mit Inselbezeichnern, Orte mit Ortsadjektiv und Städtebezeichnern, sowie Verben zu Namen können aus Platzgründen hier nicht diskutiert werden, lieferten jedoch auch Daten brauchbarer Qualität.

---

[3] Lies: Wenn zwei großgeschriebene Wörter hintereinander, danach ein kleingeschriebenes Wort vorkommt, so ist das erste Wort ein Vorname.

Der zugrunde liegende Korpus ist der des Projekts Deutscher Wortschatz (http://www.wortschatz.uni-leipzig.de) mit 36 Millionen Beispielsätzen und über 6 Millionen Wortformen für das Deutsche.

Gesucht wurden Elemente der Personennamen-Relation, sowie Titel und Berufsbezeichnungen, ein prototypischer Pattern ist "TITEL VORNAME NACHNAME".

Als Grundwissen und gleichzeitig als Startitems wurden lediglich 19 häufige Vor- und Nachnamen benutzt: Schmidt, Reuter, Wagner, Schäuble, Vogts, Hoffmann, Schulz, Möller, Maier, Beck, Michael, Thomas, Klaus, Wolfgang, Hans, Werner, Martin, Walter und Karl. Des Weiteren wurden zwölf reguläre Ausdrücke (wie z.B. *inister, *räsident) zum Taggen von Titeln verwendet und es wurde ein Längenfilter für Vornamen implementiert (Vornamen haben normalerweise 4-10 Zeichen).

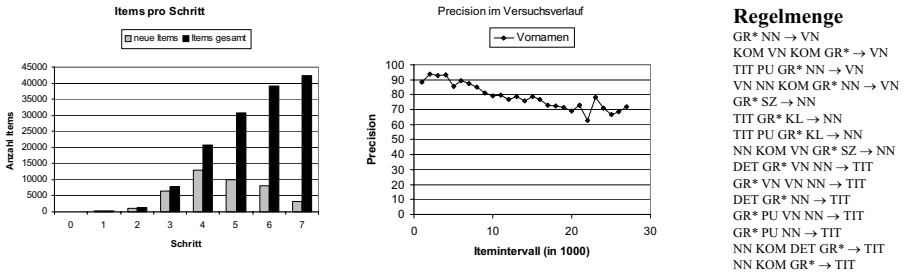

Abb. 4: Items pro Schritt, Precision im Versuchsverlauf für Vornamen und Regelmenge. Tags: GR=Groß. KL=klein, KOM=Komma, PU=Punkt, DET=Artikel.

Trotz des kleinen Grundwissens setzte der Pendelprozess ein und fand in den ersten sieben Schritten über 42000 Items, davon 15,0% Vornamen, 71,0% Nachnamen und 11,0% Titel. Der Versuch verlief gemäß dem logistischen Wachstum, siehe vorstehende Abbildung links.

Die Precision für Nachnamen und Titel erreichte über 99%, bei den Vornamen fiel die Precision während des Versuchsverlaufes leicht ab, siehe Abb. 4 Mitte.

Versuche mit italienischen und englischen Korpora zeigten ein ähnliches Bild. Hier macht sich die Sprachunabhängigkeit des Verfahrens bemerkbar, die so weit geht, dass mit dem für das Deutsche erstellte Grundwissen und den deutschen Regeln vergleichbare Ergebnisse in anderen Sprachen erzielt wurden. Die Fehlerraten waren aufgrund der Kleinschreibung für Substantive sogar geringer (Precision im Englischen: 92,6% für Vornamen, Nachnamen über 99%).

Auffallend war weiterhin, dass der Algorithmus aufgrund des deutschen Grundwissens zunächst nur Items aus der Subgruppe der deutschen Namen innerhalb der englischen (bzw. italienischen) Texte fand, und erst nach und nach zu englischen (bzw. italienischen) Namen überging.

Die nächste Abbildung zeigt Messwerte für einen Versuch mit automatisch erzeugter Regelmenge, wobei der Schwellwert für die Regelkonfidenz auf 0,7 gesetzt wurde. Ein Trainingstext von 236 Sätzen lieferte insgesamt 177 Regeln.

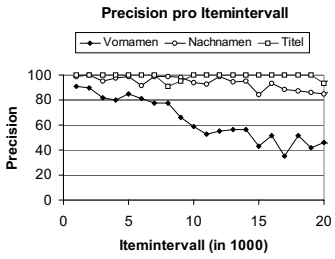

Abb. 5: Precision im Verlauf bei automatisch inferierten Regeln.

Der Hauptanteil falscher Vornamen bestand aus großgeschriebenen englischen Wörtern jeder Art, wie z.B. Let, Change, Rule, For, Secret. Verantwortlich dafür ist beispielsweise eine gewisse "Bibliothekarin Mary Love", deren Nachnamen in Songtiteln wie "Let Love Rule" vorkommt. Derartige Titel werden im Text meist in Anführungszeichen aufgeführt, so dass sie mit Hilfe eines Tagsets, das dieses Feature abbildet, ausgeschlossen werden können. Ein flexiblerer Ansatz wäre das Regellernen während des Pendelprozesses, siehe unten.

4 Ausblick

4.1 Regellernen während des Pendel-Prozesses

In der vorliegenden Implementierung geschieht das Regellernen vor dem eigentlichen Pendelprozess und benötigt einen Trainingstext, für den ein ungleich größerer Aufwand getrieben werden muss als zum Erstellen von manuellen Regeln und den Startitems-Listen. Wie weitere Versuche zeigten, sind insbesondere die Regeln für die Konvergenz des Verfahrens verantwortlich, da bei zu unscharfen Regeln wegen der Fehlerfortpflanzung die Menge der gefundenen Items im Verhältnis zu den tatsächlich gesuchten Items unverhältnismäßig groß wird.

In einer zukünftigen Version sollen aus diesem Grund Regeln auch während des Pendel-Prozesses gelernt werden. So wird ein Bootstrapping-Verfahren mit zwei Ebenen implementiert: Ausgehend von einigen Startitems und einfachen, manuell erstellten Regeln werden mit Hilfe der Regeln neue Items gesucht und mit Hilfe der Items

neue Regeln konstruiert. Die neu konstruierten Regeln befinden sich zunächst in einer Testphase und werden erst zur Klassifikation von Items verwendet, wenn sie genügend häufig mit ausreichender Genauigkeit Items richtig klassifiziert haben.

Vergleichbar damit ist (Yangarber 2002): Hier wird zum Zwecke des Erkennens negativer Belegstellen eine weitere Itemklasse für falsche Items eingeführt und ebenso wie die Zielklassen durch Bootstrapping erweitert. Außerdem erhalten hier auch die Items ein Rating.

4.2 Supervisor-Schritt zum Ablehnen von Items

Des Öfteren werden durch wenige falsche Items zahlreiche neue falsche Items gefunden, wodurch die Fehlerrate steigt und die Ergebnisse in Rohform für mögliche Anwendungen unbrauchbar macht. Abhilfe könnte an dieser Stelle eine Abkehr vom rein unüberwachten Lernparadigma schaffen: Für jedes Item wird gespeichert, aufgrund welcher anderer Items es im Verifikationsschritt akzeptiert wurde. Wird nun ein Item durch die menschliche Überwachungsinstanz als falsch eingeordnet, werden die Akzeptanzraten der mit Hilfe dieses Items klassifizierten anderen Items neu berechnet. Fällt hierbei ein anderes Item unter die Akzeptanzschwelle, so wird es ebenfalls als falsch markiert, dieser Prozess wird iteriert. Auf diese Weise kann die Qualität der aus dem Pendel-Verfahren gewonnenen Daten schnell verbessert werden.

Literatur

Bartschat, B. (1996): Methoden der Sprachwissenschaft, Erich Schmidt Verlag, Berlin.

Bohnet, B., Klatt, S., Wanner, L. (2002): A Bootstrapping Approach to Automatic Annotation of Functional Information of Adjectives with an Application to German, Proceedings of the LREC-3 Workshop: Linguistic Knowledge Acquisition and Representation - Bootstrapping Annotated Language Data, Las Palmas, Spanien.

Brin, S. (1998): Extracting Patterns and Relations from the World Wide Web, in Proceedings of the WebDB Workshop (EDBT-98), Valencia, Spanien.

Califf, M. E., Mooney, R. J. (1997): Relational Learning of Pattern-Match Rules for Information Extraction, Proceedings of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing.

Church, K. W. (1988): A stochastic parts program and noun phrase parser for unrestricted text, Proceedings of the ANLP 2.

Collins, M., Singer, Y. (1999): Unsupervised Models For Named Entity Classification. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

Duclaye, F., Yvon, F. und Collin, O. (2002): Using the Web as a Linguistic Resource for Learning Reformulations Automatically, Proceedings of the Third Conference on Language Resources and Evaluation (LREC), Las Palmas, Spanien.

Grishman, R. (1995): Where's the Syntax? The NYU MUC-6 System, Proceedings of the Sixth Message Understanding Conference, Morgan Kaufmann Publishers, San Francisco.

Manning, C. D., Schütze, H. (1999): Foundations of Statistical Natural Language Processing, Massachusetts Institute of Technology.

McEnery, T., Wilson, A. (1996): Corpus Linguistics, Edinburgh University Press.

Quasthoff, U. (1998): Projekt Der Deutsche Wortschatz, in Heyer, G. / Wolff, C. (Hrsg.), Linguistik und neue Medien, Deutscher Universitätsverlag, Wiesbaden.

Quasthoff, U., Biemann, C., Wolff, C. (2002): Named entity learning and verification: EM in large corpora, Proceedings of the Sixth Conference on Natural Language Learning (CoNNL-2002), Morgan Kaufmann Publishers, San Francisco.

Riloff, E., Jones, R. (1999): Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. Proceedings of the sixteenth National Conference on Artificial Intelligence (AAAI-99).

Yangarber, R., Lin, W., Grishman, R. (2002): Unsupervized Learning of Generalized Names, Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002), Taipei, Taiwan.