Dokumentation

Thema

Extraktion von SPO-Tripeln aus englischen Beispielsätzen

Motivation

Bei einem SPO-Tripel handelt es sich um ein Tripel bestehend aus einem **S**ubjekt, einem **P**rädikat und einem **O**bjekt. Diese können aus Sätzen extrahiert werden um deren Inhalt auf die Kernaussage zu reduzieren, da z.B. Adjektive bei der Extrakton entfernt werden.

Aufgabenstellung

Prototypische Implementierung eines Tools, das mit Hilfe der Stanford NLP Software[1] SPO-Tripel aus Beispiel-Texten extrahiert.

SPO-Tripel

Wie eingangs erwähnt handelt es sich bei SPO-Tripeln um 3-Tupel bestehend aus **S**ubjekt, **P**rädikat und **O**bjekt.

Das Subjekt in einem einfachen Englischen Satz ist die Person oder die Sache über die eine Aussage getätigt wird, wie z.B. in

John runs.

John is a teacher.

oder

John was hit by a car.

In diesen Fällen ist jeweils *John* das Subjekt.

In der traditionellen englischen Grammatik ist das Prädikat eines der beiden Hauptbestandteile eines Satzes. Der andere Hauptbestandteil ist das Subjekt. Der Zweck des Prädikats ist es, die Vorstellung über ein Subjekt abzuschließen, indem es aussagt was jenes macht oder wie es ist. Im Satz

Ben reads the book.

ist nach der traditionellen Grammatik *reads the book* das Prädikat.

In der modernen englischen Grammatik entspricht das Prädikat weitestgehend dem Hauptverb und den Hilfsmitteln welches dieses mit sich bringt. Dabei sind die

[1] http://nlp.stanford.edu

Argumente, die den Sinn des Prädikats ausmachen, nicht Bestandteil des Prädikats. Zu diesen Argumenten zählen z.B. Subjekt- und Objektnominalphrasen.

Im Beispielsatz von zuvor

Ben reads the book.

ist nach der modernen englischen Grammatik lediglich *reads* das Prädikat, *Ben* und *the book* dessen Argumente. Diese Relation kann in folgender Form dargestellt werden:

reads(Ben, the book)

Hierbei steht das Prädikat außerhalb und dessen Argumente innerhalb der Klammern.

In der traditionellen englischen Grammatik wird das Objekt als der Teil des Satzes verstanden, auf den sich die Handlung des Subjekts bezieht. Es findet also eine grundlegende Unterscheidung zwischen dem Subjekt und dem Objekt in Bezug auf die Handlung statt, die vom Verb ausgedrückt wird. *The book,* im vorangegangenen Beispiel, ist also das Objekt des Satzes.

Es muss jedoch nicht unbedingt ein Objekt in jedem Satz geben. Beispiel:

They lie often.

Subjekt und Prädikat mit *They* und *lie* sind hier leicht zu bestimmen, doch *often* ist lediglich ein Adverb, welches die Handlung näher beschreibt.

In der traditionellen Grammatik wäre das Objekt ein Teil des Prädikats, in der modernen Grammatik ist es ein Argument dessen, wie das Subjekt.

Da es in dieser Arbeit darum geht Tripel aus Subjekt, Prädikat und Objekt zu finden, wäre die traditionelle Sichtweise der Grammatik nicht hilfreich, da dabei das Objekt Bestandteil des Prädikats wäre. Aus diesem Grund wird die moderne Auffassung der Grammatik angewendet, da diese auch die Möglichkeit eröffnet einen Satz auf dessen Kernaussage zu reduzieren.

Stanford CoreNLP

Bei der verwendeten Software Standford CoreNLP[2] (NLP = **N**atural **L**anguage **P**rocessing) handelt es sich um eine Sammlung von Werkzeugen zur Verarbeitung von menschlicher Sprache. Es bietet unter anderem die Möglichkeit Wörter auf ihre Grundform zu reduzieren, deren Wortarten zu bestimmen, die Normalisierung von Daten, Zeiten und numerischen Größen, die Struktur eines Satzes in Bezug auf Phrasen und syntaktische Abhängigkeiten zu identifizieren, anzugeben welche Nominalphrase sich auf die selbe Entität bezieht, uvm.

[2] https://stanfordnlp.github.io/CoreNLP/

Stanford CoreNLP enthält viele Werkzeuge der NLP Gruppe, wie zum Beispiel den part-of-speech (POS) tagger zum Bestimmen von Wortarten, den named entity recognizer (NER) zum Bestimmen von Personen-, Unternehmens- und Ortsangaben.

POS-Tagging

Unter Part-of-speech Tagging versteht man die Zuordnung von Wörtern und Satzzeichen eines Textes zu Wortarten, wie zum Beispiel Substantiv, Verb oder Adjektiv. Hierzu wird sowohl die Definition des Wortes als auch der Kontext (z. B. angrenzende Adjektive oder Nomen) berücksichtigt.

Der in Stanford CoreNLP enthaltene POS-Tagger nutzt für die englische Sprache die Tags der Penn Treebank[3]. Beispiel:

John was the CEO of a company.

POS-Tags:

```
[NNP, VBD, DT, NN, IN, DT, NN, .]
```

In Rot sind die Substantive hervorgehoben. Ohne eine Ansammlung von Regeln ist es uns nicht möglich aus den erhaltenen POS-Tags programmatisch ein SPO-Tripel zu generieren. Diese Regeln würden dazu dienen das Subjekt aus den Substantiven zu bestimmen, indem angrenzende Wortarten und die Position des Substantivs im Satz einbezogen würde. Bei diesem Ansatz wäre die NLP-Software aus Standfort nur eine geringe Hilfe und die Bestimmung sowie Anwendung der Regeln wäre ein aufwändiges Unterfangen.

Abhängigkeitsbäume

Die Stanford CoreNLP-Softwaresammlung bietet jedoch noch mehr Möglichkeiten Wörter in Sätzen einzuordnen. Und zwar mit Abhängigkeiten zueinander.

Die sogenannten *Stanford Dependencies*[4] oder seit Version 3.5.2 von Stanford CoreNLP *Universal Dependencies*[5] bieten eine Darstellung von grammatikalischen Beziehungen zwischen Wörtern innerhalb eines Satzes. Dabei handelt es sich um Triplets bestehend aus dem Namen der Relation, der Quelle und dem Ziel der Beziehung.

Nehmen wir erneut das einfache Beispiel von zuvor:

Ben reads the book.

Abhängigkeitsbaum:

```
nsubj(reads/VBZ,Ben/NNP)
det(book/NN,the/DT)
dobj(reads/VBZ,book/NN)
punct(reads/VBZ,./.)
```

[3] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
[4] https://nlp.stanford.edu/software/dependencies_manual.pdf
[5] http://universaldependencies.org/

Auf den ersten Blick fällt auf, dass zusätzlich zu Wörtern auch deren POS-Tags annotiert sind. Diese können gemeinsam mit den Relationen hilfreich zur Bestimmung der SPO-Tripel sein.

Die Bestimmung des Subjekts ist hier einfach, da es sich bei `nsubj` um ein nominal subject[6] handelt kann das Ziel der Relation `Ben` als Subjekt übernommen werden. Die Quelle der Relation `reads` ist weder ein Substantiv, Adjektiv oder Adverb, weswegen es sich ziemlich sicher um das Prädikat des Satzes handeln muss. Hier kommt die Zuhilfenahme der POS-Tags ins Spiel. Würde es sich nämlich um ein Substantiv handeln, dann wäre es das Objekt des Satzes, wie folgendes Beispiel zeigt:

John was the CEO of a company.

Abhängigkeitsbaum:

```
nsubj(CEO/NN,John/NNP)
cop(CEO/NN,was/VBD)
det(CEO/NN,the/DT)
case(company/NN,of/IN)
det(company/NN,a/DT)
nmod:of(CEO/NN,company/NN)
punct(CEO/NN,./.)
```

In diesem Beispiel ist `John` das Subjekt und ebenso durch `nsubj` identifiziert worden. Da es sich hier bei der Quelle der Relation `CEO` um ein Substantiv handelt, kann diese nicht das Prädikat des Satzes sein. Es ist das Objekt und kann in diesem Fall als solches übernommen werden.

Doch zunächst zurück zu Ben, in diesem Beispiel gilt es noch das Objekt zu bestimmen. Im Abhängigkeitsbaum wird dieses durch die Relation `dobj`[7] gekennzeichnet. Diese beinhaltet als Ziel das Objekt `book` und als Quelle das Prädikat `reads` welches allerdings schon zuvor als solches bestimmt wurde.

Als Ausgabe erhalten wir also:

```
Ben,read,the book
```

Im Beispiel von John fehlt noch das Prädikat, welches in der Relation `cop`[8] zu finden ist. Hierbei ist das Ziel der Relation `was` unser gesuchtes Prädikat und die Quelle `CEO` das bereits bestimmte Objekt.

Als Ausgabe erhalten wir also:

```
John,be,the CEO
```

Um die SPO-Tripel eindeutig zu bestimmen reichen also weder die POS-Tags, noch die Abhängigkeitsbäume alleine aus. Die Bäume enthalten allerdings schon aussagekräftige Informationen, die mit Hilfe der POS-Tags und der Anwendung weniger Regeln zu guten Ergebnissen führen.

---

[6] http://universaldependencies.org/u/dep/nsubj.html
[7] http://universaldependencies.org/docs/en/dep/dobj.html
[8] http://universaldependencies.org/u/dep/cop.html

Neben Nomen als Subjekt werden auch Teilsätze, die durch die Relation `csubj` und `csubjpass` im passiven Fall bestimmt werden unterstützt. Beispiel:

To read is easier than to write.

Abhängigkeitsbaum:

```
mark(read/VB,To/TO)
csubj(easier/JJR,read/VB)
cop(easier/JJR,is/VBZ)
mark(write/VB,than/IN)
mark(write/VB,to/TO)
ccomp(easier/JJR,write/VB)
punct(easier/JJR,./.)
```

Ergebnisausgabe:

```
read,be,
```

Passiv

Auch ist es mit Hilfe der Abhängigkeitsbäume möglich Passivsätze zu erkennen und daraus SPO-Tripel zu bestimmen. Beispiel:

Dole was defeated by John.

Abhängigkeitsbaum:

```
nsubjpass(defeated/VBN,Dole/NNP)
auxpass(defeated/VBN,was/VBD)
case(John/NNP,by/IN)
nmod:agent(defeated/VBN,John/NNP)
punct(defeated/VBN,./.)
```

Durch die Relation `nsubjpass`[9] anstatt `nsubj` wird hier der passive Satz identifiziert. `Dole` wird hier nicht als Subjekt übernommen, sondern als Objekt. Über die Relation `nmod`[10] kann hier das Subjekt `John` extrahiert werden.

Ergebnisausgabe:

```
John,defeat,Dole
```

Bestimmungswörter

Zum leichteren Verständnis wurde bei der Extraktion des Subjekts und Objekts geprüft, ob Wörter wie `a` oder `the` in Verbindung zu diesen stehen. Die Relation hierfür heißt det[11] und enthält die Verbindung zwischen einem Substantiv und dessen Bestimmungswort.

[9] http://universaldependencies.org/docs/en/dep/nsubjpass.html
[10] http://universaldependencies.org/en/dep/nmod.html
[11] http://universaldependencies.org/u/dep/det.html

Wortstammreduktion

Zum leichteren Verständnis und zur Vereinheitlichung der Ergebnisausgabe werden alle gefundenen Verben auf ihre Grundform reduziert. Beispiel:

John was the CEO of a company.

Ergebnisausgabe:

```
John,be,the CEO
```

Hierbei wurde das flektierte Wort `was` auf die Grundform `be` reduziert.

Negation

Ein Prädikat kann zusätzlich zur normalen Form, auch negiert auftreten. Beispiel:

Dole wasn't defeated by Clinton.

Abhängigkeitsbaum:

```
nsubjpass(defeated/VBN,Dole/NNP)
auxpass(defeated/VBN,was/VBD)
neg(defeated/VBN,n't/RB)
case(Clinton/NNP,by/IN)
nmod:agent(defeated/VBN,Clinton/NNP)
punct(defeated/VBN,./.)
```

Ergebnisausgabe:

```
Clinton,¬defeat,Dole
```

Hierbei fällt die Negation im ersten Satz auf. Im Abhängigkeitsbaum wird diese durch die Abhängigkeit `neg`[12] ausgedrückt und gibt eindeutig Auskunft über Verneinungen im Satz.

In der Ergebnisausgabe wird die Verneinung durch ein Negationszeichen (¬) deutlich gemacht.

Zukünftige Arbeiten

Bei der Implementierung handelt es sich um einen Prototyp, der vor allem aus einfachen englischen Sätzen SPO-Tripel extrahieren kann. Im fachlichen Kontext oder bei besonders komplexen Sätzen wird die Bestimmung der Relationen ungenauer und es kann dazu kommen, dass zu wenig oder falsche Tripel bestimmt werden. Zukünftig könnte, soweit möglich, daran gearbeitet werden weitere Regeln bei der Verarbeitung der Sätze anzuwenden um die Bestimmung zu verbessern.

Bei Subjekten aus Teilsätzen müssen eventuell noch weitere Bestandteile des Teilsatzes als Subjekt übernommen werden, damit die Verständlichkeit nicht verloren geht.

[12] http://universaldependencies.org/en/dep/neg.html

Quellcode

Der Quellcode kann unter https://github.com/OSwoboda/spotripel gefunden werden. Es handelt sich dabei um ein Maven-Projekt und benötigt mindestens Java 8.