Exposé zur Studienarbeit

Evaluierung von Constituent-Parsern anhand von Dependency Graphen

Stefan Pietschmann

18. Dezember 2007

Betreuer: Prof. Dr. Ulf Leser

1 Hintergrund

In der Biomedizin ist man daran interessiert, aus einem Korpus von Fachtexten nur genau die Texte oder Textstellen geliefert zu bekommen, die Informationen zu einem bestimmten Sachverhalt enthalten. Diese Sachverhalte können beispielsweise Fragen nach Protein-Protein-Interaktionen, Protein-Funktions-Zusammenhängen oder Gen-Krankheits-Beziehungen sein.

Für solche Arten der Informationsextraktion kommen verschiedene Methoden des Natural Language Processing (NLP) in Frage, wie z. B. Kookurrenz-, Pattern-Matching- oder Parsing-Verfahren. Der Vorteil von letzteren ist, dass sie die tiefere grammatikalische Satzstruktur berücksichtigen und somit bestimmte Beziehungen und Abhängigkeiten zwischen Wörtern finden können, bei dem erstere beiden Verfahren versagen, da sie nur direkt auf der Satzoberfläche arbeiten.

Bisher für solche Zwecke eingesetzte Parser sind hauptsächlich Constituent Parser und Dependency Parser. Wobei ein Constituent Parser so arbeitet, dass er einen gegebenen Satz rekursiv in seine linguistischen Komponenten (Constituents) splittet und als Output einen Syntax-Baum (Constituent Tree) liefert, bei dem die Wurzel den kompletten Satz repräsentiert, die Nicht-Blatt-Knoten die Constituents und die Blätter die einzelnen Wörter des Satzes. Ein Dependency Parser wiederum gibt auf Eingabe eines Satzes einen Graphen aus, bei dem jeder Knoten ein Wort des Satzes repräsentiert und die Kanten

die Abhängigkeiten zwischen den Wörtern darstellen. Enthalten Dependency Graphen zwar offensichtlich nicht soviel linguistische Informationen über die tiefere Satzstruktur wie Constituent Trees, sind sie dafür aber beliebter im NLP, da ihre Form näher an der logischen Prädikatenform ist, auf die man dort gern einen Satz reduzieren möchte und weil eine Tiefenstruktur, wie sie die Constituent Trees darstellen, dort schlicht nicht benötigt wird [1, S. 430]. Allerdings liegen die meisten annotierten Korpora in Form von Constituent Trees vor. Zudem gibt es dort einen de facto Standard in der Benennung der Constituents, nämlich den der *Penn Treebank* [2]. Solch ein Standard existiert für die Benennung der Dependencies bei Dependency Parsern bisher nicht.

2 Ziel

Ziel der Arbeit ist die Evaluation von zwei verschiedenen Constituent Parsern auf Basis von Dependency Graphen. Es werden also beide eingangs beschriebenen gängigen Parsing-Verfahren kombiniert. Dabei werden auf einem gegebenen Korpus zwei Constituent Parser angewendet und deren als Output entstehende Constituent Trees in Dependency Graphen transformiert. Das Korpus fungiert hier als Goldstandard und muss deshalb bereits hand-annotiert in Form von Constitutent Trees vorliegen, die wiederum ebenfalls in Dependency Graphen transformiert werden. Die Evaluation erfolgt dann anhand der Dependency Graphen der Constituent Parser bezüglich der Dependency Graphen des Goldstandards.

Dieses kombinierte Vorgehen hat den Vorteil, dass sowohl die Sätze des Korpus' bezüglich ihrer tiefen linguistischen Struktur analysiert werden, als auch, dass die Evaluation auf Ebene der – für das NLP wünschenswerten – Dependency Graphen geschieht.

Evaluiert werden die Parser jeweils vor allem im Hinblick auf ihre generelle Genauigkeit und auf ihre Geschwindigkeiten, eventuell auch auf weitere spezielle Genauigkeiten, z. B. bei bestimmten im biomedizinischen Kontext wichtigen Präpositionen, Negationen, Konjunktionen oder Verben.

3 Vorgehensweise

Die Vorgehensweise wird sich stark an der von Clegg und Shepherd orientieren, die bereits ein Paper zu selbiger Thematik veröffentlicht haben [3]. Das heißt, es müssen zunächst ein Korpus, zwei Constituent Parser und ein Tool, welches Constituent Trees nach Dependency Graphen transformiert, festgelegt werden. Alle Komponenten müssen

frei zugänglich sein (beispielsweise Open Source). Das Korpus muss zudem in annotierter Form von Constituent Trees vorliegen. Hier bietet sich ein annotierter Auszug des *GENIA-Korpus* [4] an, welches eine Menge von Abstracts des *MEDLINE Journals* umfasst (Ein Alternativ-Korpus wäre beispielsweise die *Penn Treebank*). Als zu untersuchende Constituent Parser eignen sich der *Charniak-Lease-Parser* [5][6] und der *Bikel-Parser* [7][8]. Beide wurden bereits im biomedizinischen Kontext eingesetzt. Als Tool für die Transformation kommt das *Stanford-Tool* [9] in Frage. Der Ablauf wird sich dann in folgende konkrete Schritte gliedern:

Korpus und Parser vorbereiten Bevor die Parser auf das Korpus angewendet werden können, müssen alle Komponenten vorbereitet werden

- Das GENIA-Korpus muss von seinen Tree-Annotationen wieder befreit werden (denn diese sollen die Parser ja jeweils selbst erstellen).
- Der Bikel-Parser muss trainiert werden. Dies geschieht beispielsweise anhand der Penn Treebank. Alternativ kann man auch bereits fertige Trainingsdaten nutzen, die anhand der Penn Treebank erstellt wurden. Der Parser erwartet POS-getaggten Text, d. h. es wird ein Tag-Tool benötigt. Hierfür bietet sich der *MedPost Tagger* [10] an, welcher auf einer Menge von MEDLINE Abstracts trainiert wurde.
- Der Charniak-Lease-Parser muss ebenfalls trainiert werden oder man nutzt auch hier bereits anhand der Penn Treebank erstellte zur Verfügung gestellte Daten. Ein externer POS-Tagger wird hier nicht benötigt, da der Parser über ein internes POS-Tag-Modul verfügt (trainiert auf Abstracts der GENIA-Treebank) und somit das Text-Tagging intern ausführt.

Korpus parsen Beide Parser werden auf das Korpus angewendet, so dass eine Menge von Constituent Trees entsteht. Eventuell ist eine nachträgliche Bearbeitung der Bäume der Parser und des Goldstandards nötig um ein einheitliches Constituent Tree-Format zu bekommen.

Transformation der Constituent Trees in Dependency Graphen Die Constituent Trees der Parser und des Goldstandards werden mit Hilfe des Stanford-Tools in Dependency Graphen transformiert.

Evaluation Die Evaluation erfolgt anhand des satzweisen Vergleichs der Dependency Graphen vom Goldstandard und der der Parser. Als Vergleichsmittel bieten sich Precision und Recall an bzw. ihre Kombination als F-Measure. Es muss dann

festgelegt werden, was ein True Positive ist. Hier käme die Interpretation in Frage, dass ein True Positive genau dann vorliegt, wenn in zwei Dependency Graphen eine Kante dieselbe Beschriftung, denselben Startknoten und denselben Endknoten hat.

Literatur

[1] Manning, Christopher D. ; Schütze, Hinrich: *Foundations of Statistical Natural Language Processing*. Second Printing. Cambridge/London : The MIT Press, 2000

[2] *The Penn Treebank Project*. http://www.cis.upenn.edu/~treebank/, Abruf: 18. Dezember 2007

[3] Clegg, Andrew B. ; Shepherd, Adrian J.: Benchmarking natural-language parsers for biological applications using dependency graphs. In: *BMC Bioinformatics* (2007). http://www.biomedcentral.com/1471-2105/8/24, Abruf: 18. Dezember 2007

[4] *GENIA Treebank*. http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/topics/Corpus/GTB.html, Abruf: 18. Dezember 2007

[5] Lease, Matthew ; Charniak, Eugene: Parsing Biomedical Literature. In: *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'05)* (2005), 58-69. http://citeseer.ist.psu.edu/753793.html, Abruf: 18. Dezember 2007

[6] *Charniak-Lease-Parser*. http://bllip.cs.brown.edu/resources.shtml, Abruf: 18. Dezember 2007

[7] Bikel, Dan: Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In: *Proceedings of the Human Language Technology Conference 2002 (HLT2002)* (2002). http://www.cis.upenn.edu/~dbikel/#research, Abruf: 18. Dezember 2007

[8] *Bikel-Parser*. http://www.cis.upenn.edu/~dbikel/download.html, Abruf: 18. Dezember 2007

[9] *Stanford NLP Tools*. http://nlp.stanford.edu/software/index.shtml, Abruf: 18. Dezember 2007

[10] Smith, L. ; Rindflesch, T. ; Wilbur, W. J.: MedPost: a part-of-speech tagger for bioMedical text. In: *Bioinformatics* (2004). http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=PubMed&dopt=AbstractPlus&list_uids=15073016, Abruf: 18. Dezember 2007