Inkrementelles Lernen latenter linguistischer Strukturen durch approximative Suche

Anders Björkelund[1]

Abstract: Diese Dissertation setzt maschinelles Lernen ein, um aus linguistisch annotierten Textsammlungen Modelle für die Analyse natürlicher Sprache zu trainieren (Natural Language Processing, NLP), beispielsweise für die Ermittlung der syntaktischen Struktur von Sätzen in einem Textdokument. Die Suchräume, die sich in diesem Bereich ergeben, sind in der Regel so umfangreich, dass exakte Suchverfahren nicht in Frage kommen. In der Praxis muss jeder Ansatz einen Kompromiss finden zwischen der Ausdrucksstärke der verwendbaren Features einerseits und der Effizienz des Suchverfahrens andererseits.

Die Arbeit präsentiert ein Meta-Framework, das sich für unterschiedliche Aufgaben aus der maschinellen Sprachverarbeitung instantiieren lässt und das es jeweils erlaubt, mit alternativen Strategien beim maschinellen Lernen zu experimentieren. So wird eine systematische Untersuchung von Verfahren des inkrementellen Lernens mit latenten linguistischen Strukturen und approximativen Suchverfahren ermöglicht. Es zeigt sich, dass etablierte Methoden aus der aktuellen NLP-Forschung sehr empfindlich sind, was die Wahl von Updatemethoden betrifft. Wir schlagen neue Updatemethoden vor und zeigen, dass diese zu mindestens gleichwertigen Ergebnissen führen, in einigen Fällen jedoch zu erheblichen Qualitiätsverbesserungen. Die Resultate tragen zu einem vertieften Verständnis der Charakteristika unterschiedlicher Analyseaufgaben bei.

1 Einführung

Bei der automatisierten Analyse natürlicher Sprache werden in der Regel maschinelle Lernverfahren eingesetzt, um verschiedenste linguistische Information wie beispielsweise syntaktische Strukturen vorherzusagen. **Structured Prediction** (dt. etwa Strukturvorhersage; [Sm11]), also der Zweig des maschinellen Lernens, der sich mit der Vorhersage komplexer Strukturen wie formalen Bäumen oder Graphen beschäftigt, hat deshalb erhebliche Beachtung in der Forschung zur automatischen Sprachverarbeitung gefunden.

In manchen Fällen ist es vorteilhaft, die gesuchte **linguistische Struktur** nicht direkt zu modellieren und stattdessen **interne Repräsentationen** zu lernen, aus denen dann die gewünschte linguistische Information abgeleitet werden kann. Da die internen Repräsentationen allerdings selten direkt in Trainingsdaten verfügbar sind, sondern erst aus der linguistischen Annotation inferiert werden müssen, kann es vorkommen, dass dabei mehrere äquivalente Strukturen in Frage kommen. Anstatt nun vor dem Lernen eine Struktur beliebig auszuwählen, kann man diese Entscheidung dem Lernverfahren selbst überlassen, welches dann selbständig die für das Modell am besten passende auszuwählen lernt. Unter diesen Umständen bezeichnet man die interne, nicht a priori bekannte Re-

[1] Department of Astronomy and Theoretical Physics, Lund University, anders.bjorkelund@thep.lu.se

präsentation für eine gesuchte Zielstruktur als **latent** (vgl. das Beispiel in Abb. 1 für die Aufgabe der Koreferenz-Resolution).

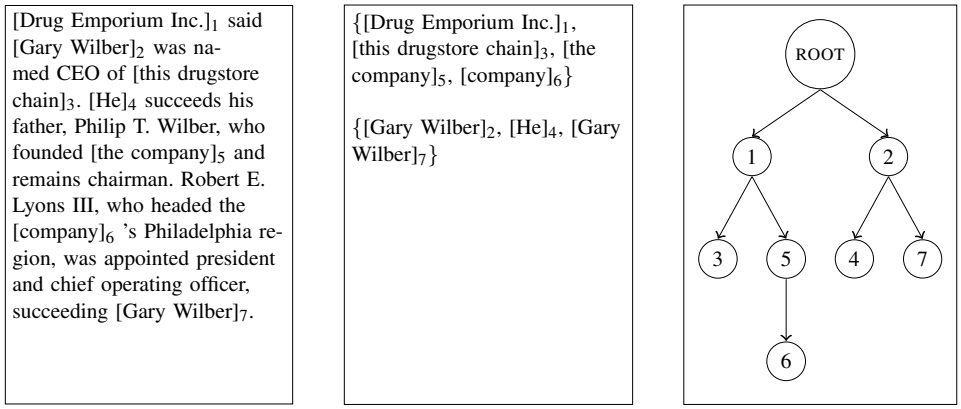


Abb. 1: *Illustration der Eingabe-, Ausgabe- und Vorhersageräume am Beispiel der Aufgabenstellung der Koreferenz-Resolution: links ist als Eingabebeispiel ein Textdokument mit markierten referentiellen Phrasen dargestellt, in der Mitte die Ausgabestruktur der Koreferenz-Resolution: eine Partition sämtlicher referentieller Phrasen in Mengen mit dem gleichen Referenten. Rechts ist eine (mögliche) latente Baumstruktur dargestellt, die sich im Zuge des Lernprozesses als effektiv erwiesen haben mag (die Entscheidung, eine gegebene Phrase der einen oder einer anderen existierenden Partition zuzuschlagen kann so auf möglichst zuverlässiger Basis erfolgen).*

Diese Dissertation stellt ein **Structured Prediction Framework** vor, mit dem man den Vorteil latenter Repräsentationen nutzen kann und welches gleichzeitig von konkreten Anwendungsfällen abstrahiert. Diese Modularisierung ermöglicht die Wiederverwendbarkeit und den Vergleich über mehrere Aufgaben und Aufgabenklassen hinweg. Um das Framework auf ein reales Problem anzuwenden, müssen nur einige Hyperparameter definiert und einige problemspezifische Funktionen implementiert werden.

Das vorgestellte Framework basiert auf dem **Structured Perceptron** [Ro58, Co02]. Der Perceptron-Algorithmus ist ein inkrementelles Lernverfahren (engl. online learning), bei dem während des Trainings einzelne Trainingsinstanzen nacheinander betrachtet werden. In jedem Schritt wird mit dem aktuellen Modell eine Vorhersage gemacht. Stimmt die Vorhersage nicht mit dem vorgegebenen Ergebnis überein, wird das Modell durch ein entsprechendes Update angepasst und mit der nächsten Trainingsinstanz fortgefahren. Das Structured Perceptron wird im vorgestellten Framework mit **Beam Search** kombiniert. Beam Search ist ein approximatives Suchverfahren, welches auch in sehr großen Suchräumen effizientes Suchen erlaubt (s. Abb. 2; hier wird *Beam Search* im Vergleich zu einem radikaleren approximativem Verfahren, *Greedy Search* illustriert). Es kann aus diesem Grund aber keine Garantie dafür bieten, dass das gefundene Ergebnis auch das optimale ist. Das Training eines Perceptrons mit Beam Search erfordert deshalb besondere Update-Methoden, z.B. *Early-* [CR04] oder *Max-Violation-Updates* [HFG12], um mögliche Vorhersagefehler, die auf den Suchalgorithmus zurückgehen, auszugleichen.

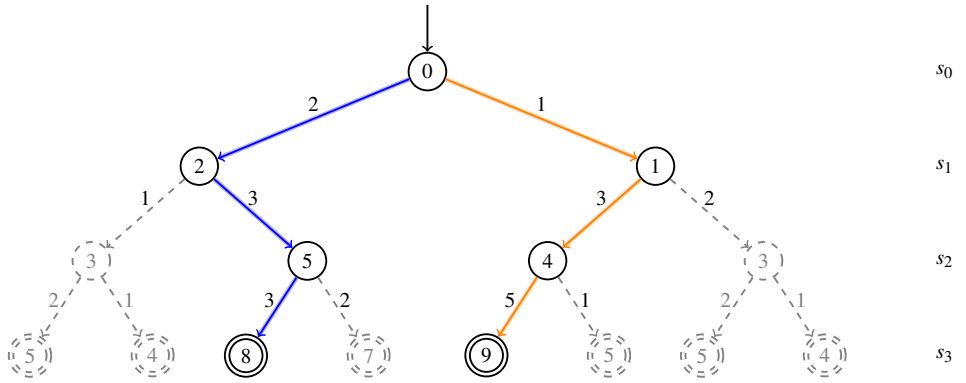


Abb. 2: *Beispiel eines Suchbaums (mit einer positiven* Score*-Wertung an den Kanten; gesucht ist der Pfad mit der größten Summe); in diesem Beispiel führt eine* Greedy Search*-Strategie zu einem suboptimalen Ergebnis (blauer Pfad), da der optimale (orange-farbene) Pfad erst spät zu einer höheren Wertung führt. Ein* Beam Search*-Verfahren mit* Beam*-Weite 2 würde die beiden farbig hervorgehobenen Pfade berücksichtigen (und die gestrichelten Pfade verwerfen).*

Das Framework ist angelegt auf einen systematischen Vergleich des Effekts verschiedener *Machine Learning*-Strategien: approximative Suchverfahren (mit **Beam Search** oder **Greedy Search**) werden gegenübergestellt mit exakten Suchverfahren (mit eingeschränkter Ausdrucksstärke der einsetzbaren Features); für approximative Suchverfahren können unterschiedliche Update-Methoden verglichen werden. Über die in der Literatur vorgeschlagenen Methoden hinaus werden weitere Techniken entwickelt, insbesondere die Erweiterung der **LaSO**-Methode ("Learning as Search Optimization") von [DIM05] als "Delayed LaSO" (**DLaSO**). Abb. 3 illustriert den Effekt unterschiedlicher Update-Methoden anhand eines abstrakten Suchbaums.

2 Behandelte NLP-Strukturvorhersageaufgaben

Das Framework wird in der Dissertation auf drei NLP-Aufgaben angewandt, für die ein überwachtes Lernen jeweils mit einer latenten Vorhersagerepräsentation umgesetzt werden kann: Koreferenzresolution [BK14], Dependenzparsing [BN15] und Dependenzparsing mit gleichzeitiger Satzsegmentierung [Bj16].

Das vorgestellte Modell zur **Koreferenzresolution** ist eine Erweiterung eines existierenden Modells [FdSM12], welches Koreferenz mit Hilfe latenter Baumstrukturen repräsentiert (wie oben bereits in Abb. 1 illustriert). Dieses Modell wird um Features erweitert, mit denen nicht-lokale Abhängigkeiten innerhalb eines größeren strukturellen Kontexts modelliert werden. Die Modellierung nicht-lokaler Abhängigkeiten macht durch die kombinatorische Explosion der Features die Verwendung eines approximativen Suchverfahrens notwendig. Es zeigt sich aber, dass das so entstandene Koreferenzmodell trotz der approximativen Suche dem Modell ohne nicht-lokale Features überlegen ist, sofern hinreichend gute Update-Verfahren beim Lernen verwendet werden. Für das **Dependenzparsing**

verwenden wir ein transitionsbasiertes Verfahren, bei dem Dependenzbäume (Abb. 4) inkrementell durch Transitionen zwischen definierten Zuständen konstruiert werden [Ni08]. Im ersten Schritt erarbeiten wir eine umfassende Analyse des latenten Strukturraums eines bekannten Transitionssystems, nämlich ArcStandard mit Swap [Ni09]. Diese Analyse erlaubt es uns, die Rolle der latenten Strukturen in einem transitionsbasierten Dependenzparser zu evaluieren. Wir zeigen dann empirisch, dass die Nützlichkeit latenter Strukturen von der Wahl des Suchverfahrens abhängt – in Kombination mit Greedy-Search verbessern sich die Ergebnisse, in Kombination mit Beam-Search bleiben sie gleich oder verbessern sich leicht gegenüber vergleichbaren Modellen. Für die dritte Aufgabe wird der Parser noch einmal erweitert: wir entwickeln das Transitionssystem so weiter, dass es neben **syntaktischer Struktur auch Satzgrenzen vorhersagt** (vgl. Abb. 5) und testen das System auf verrauschten und unredigierten Textdaten. Mit Hilfe sorgfältig ausgewählter Baselinemodelle und Testdaten messen wir den Einfluss syntaktischer Information auf die Vorhersagequalität von Satzgrenzen und zeigen, dass sich in Abwesenheit orthographischer Information wie Interpunktion und Groß- und Kleinschreibung das Ergebnis durch syntaktische Information verbessert.

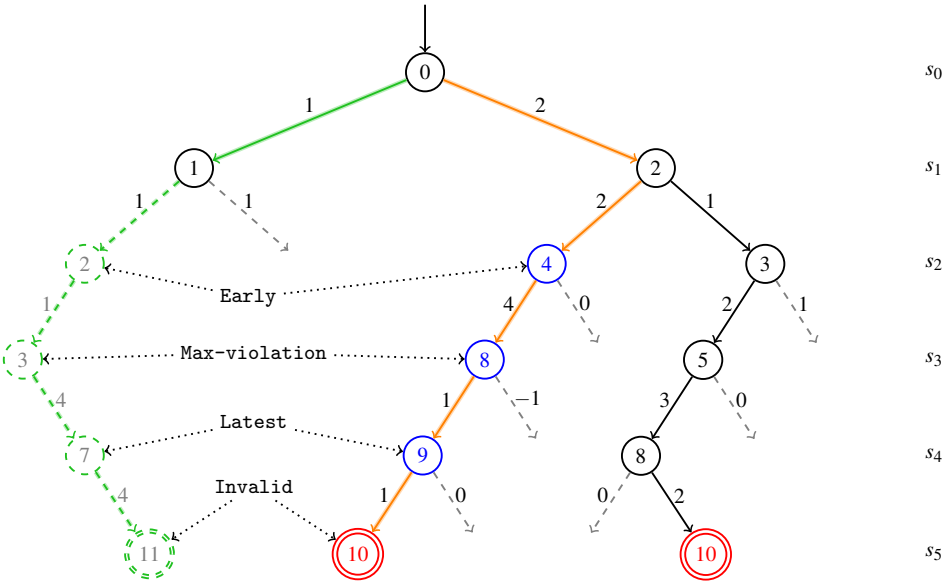


Abb. 3: *Beispiel-Illustration unterschiedlicher Update-Verfahren anhand eines Suchbaums für* Beam Search *mit* Beam-*Weite 2 (die Zustände innerhalb des* Beam*s sind mit durchgezogenen Linien dargestellt). Der korrekte Pfad ist grün hervorgehoben. Die blau hervorgehobenen Zustände sind jeweils die Basis für einen möglichen validen Update (gepaart mit den korrespondierenden grünen Zuständen außerhalb des* Beam*s). Rot hervorgehoben sind Zustände, anhand derer kein valider Update mehr möglich wäre (bei Schritt s_5 ist der Score des (verlorenen) korrekten Pfads größer als des besten* Beam-*Pfads.*

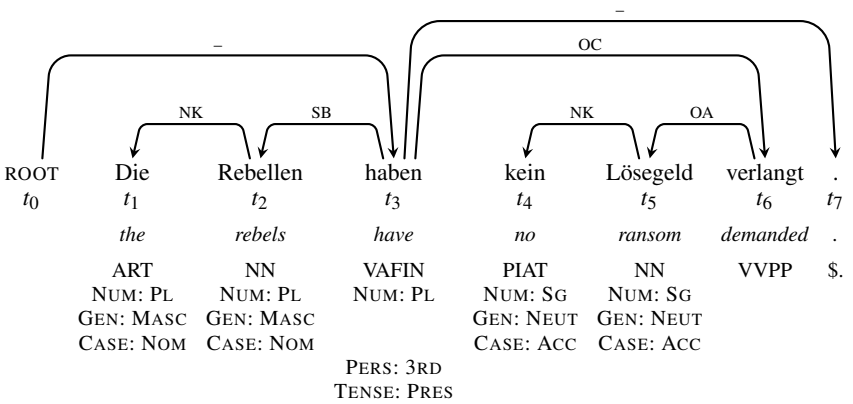
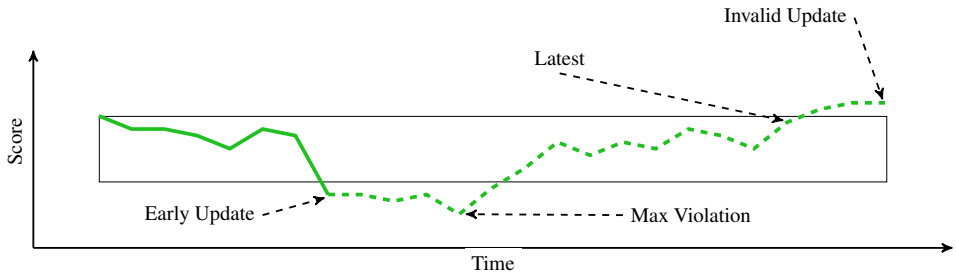


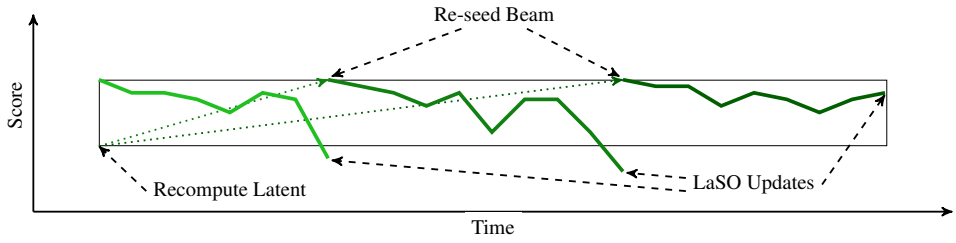
Abb. 4: *Dependenzstrukturbaum zur Illustration der Zielstruktur für die Aufgabe des Dependenzparsings. Die Baumstruktur oben entspricht der gesuchten Ausgabestruktur* **y***, die Eingabe* **x** *besteht in der Tokenfolge unten, einschl. atomaren Features auf den Tokens wie etwa deren Part-of-Speech-Tags.*

> [1]cum ergo natus esset Iesus in Bethleem Iudaeae in diebus Herodis regis ecce magi ab oriente venerunt Hierosolymam [2]dicentes ubi est qui natus est rex Iudaeorum vidimus enim stellam eius in oriente et venimus adorare eum [3]audiens autem Herodes rex turbatus est et omnis Hierosolyma cum illo [4]et congregans omnes principes sacerdotum et scribas populi sciscitabatur ab eis ubi Christus nasceretur
>
> "[1]Now when Jesus was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to Jerusalem, [2]Saying, Where is he that is born King of the Jews? for we have seen his star in the east, and are come to worship him. [3]When Herod the king had heard these things, he was troubled, and all Jerusalem with him. [4]And when he had gathered all the chief priests and scribes of the people together, he demanded of them where Christ should be born."

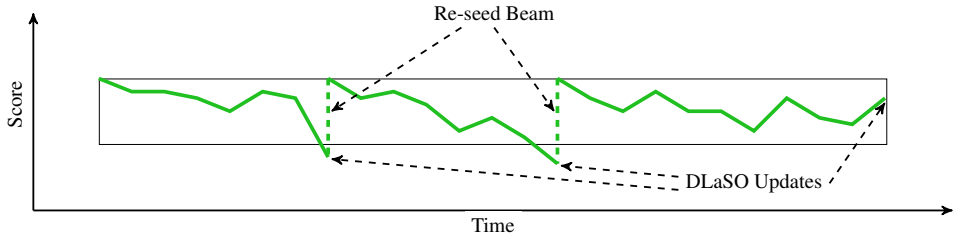
Abb. 5: *Beispiel zur Illustration der Herausforderungen bei der NLP-Aufgabe der gemeinsamen Satzgrenzenbestimmung und Satzstrukturanalyse: oben findet sich der Anfang von Matthäus 2 aus der lateinischen Vulgate-Bibel, unten die entsprechende englische Übersetzung in der King-James-Fassung. Vers-Zahlen sind als Superskripte dargestellt; satzinitiale Tokens sind unterstrichen dargestellt.*



(a) Early, Max-violation, Latest.

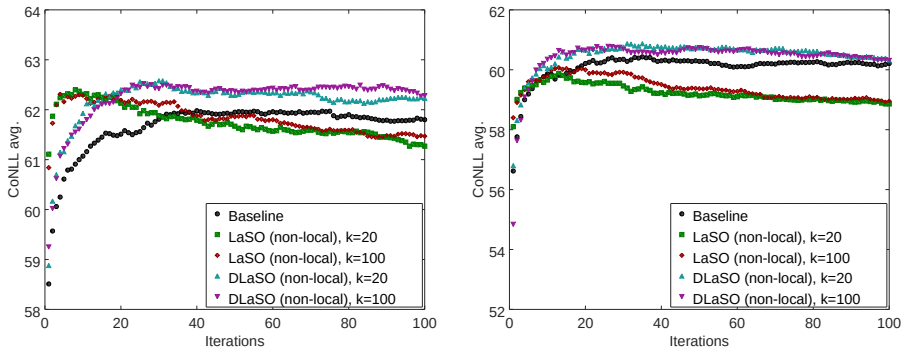


(b) LaSO.



(c) DLaSO.

Abb. 6: *Graphische Darstellung des Effekts unterschiedlicher Update-Strategien.*



(a) English (b) Chinese

Abb. 7: *Koreferenz-Resolution: Lernkurven für den Vergleich einer Baseline (exakte Suche mit ausdrucksschwächeren Features) und **LaSO** ("Learning as Search Optimization") [DIM05] und **DLaSO** ("Delayed Learning as Search Optimization"); (a) Kurve für Englisch, (b) Kurve für Chinesisch (Evaluation nach kombiniertem Maß aus CoNLL-Shared-Task zu Koreferenz-Resolution)*

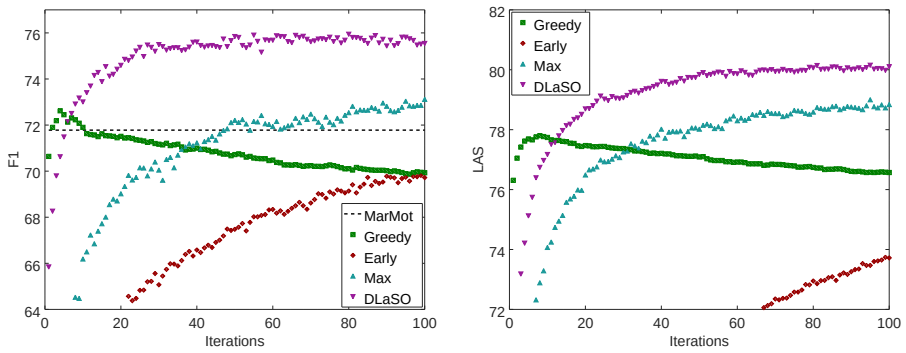
3 Verbesserte Update-Methoden bei der approximativen Suche

Das abstrakte Framework für die Modellierung von Strukturvorhersageaufgaben als Suchproblem mit einer latenten Vorhersagerepräsentation ermöglicht systematische Vergleichsexperimente zur Effektivität von unterschiedlichen *Machine Learning*-Strategien.

In der Dissertation wird das Framework für jede der drei erwähnten NLP-Analyseaufgaben instantiiert und führt einerseits zu aufgabenspezifischen empirischen Ergebnissen, andererseits zu systematischen Einsichten aus dem Vergleich. So erweist sich z.B. in mehreren Experimenten, dass die etablierten Update-Methoden, also Early- oder Max-Violation-Update, nicht mehr gut funktionieren, sobald die vorhergesagte Struktur eine gewisse Größe überschreitet.

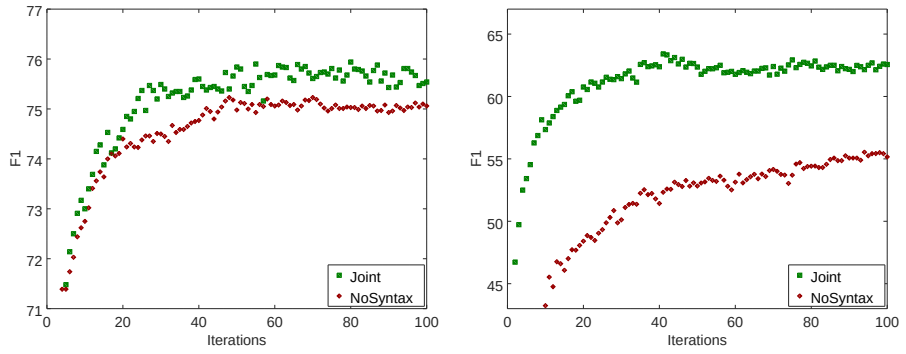
Es zeigt sich, dass das Hauptproblem dieser Methoden das **Auslassen von Trainingsdaten** ist, und dass sie desto mehr Daten auslassen, je größer die vorhergesagte Struktur wird. Dieses Problem kann durch bessere Update-Methoden vermieden werden, bei denen stets alle Trainingsdaten verwendet werden. Wir stellen eine neue Methode vor, **DLaSO** ("Delayed Learning as Search Optimization") [BK14], und zeigen, dass diese Methode konsequent bessere Ergebnisse liefert als alle Vergleichsmethoden. In der schematischen Darstellung in Abb. 6 wird deutlich, wie das Zurücksetzen der *Beam*-Kandidaten dazu führt, dass das Auslassen von Trainingsdaten vermieden wird; Abb. 7-9 zeigen Ausschnitte aus den experimentellen Ergebnissen.

Überdies zeigen die Experimente, dass eine erhöhte Beamgröße beim Suchen das Problem der ausgelassenen Trainingsdaten nicht kompensieren kann und daher keine Alternative zu besseren Update-Methoden darstellt.



(a) sentence boundary detection (b) parsing

Abb. 8: *Satzgrenzen- und Dependenzstruktur-Vorhersage mit approximativen Suchverfahren: Vergleich unterschiedlicher Update-Strategien auf den Switchboard-Korpus-Entwicklungsdaten. (a) zeigt den F_1 für Satzgrenzenerkennung; (b) Labeled Attachment Score für die Dependenzparsing-Aufgabe. DLaSO führt für beide Teilaufgabe zu einer erheblich höheren Vorhersage-Qualität.*



(a) Switchboard (b) WSJ*

Abb. 9: *Ergebnisse zur gemeinsamen Satzgrenzen- und Dependenzstruktur-Vorhersage: Effekt der Verfügbarkeit von syntaktischer Information für die Satzgrenzenvorhersage für zwei unterschiedliche Korpora: grün mit syntaktischer Information, rot ohne. (a) Switchboard; (b) WSJ* (Zeitungstext ohne Satzzeichen und ohne Großschreibung am Satzanfang).*

Literaturverzeichnis

[Bj16] Björkelund, Anders; Faleńska, Agnieszka; Seeker, Wolfgang; Kuhn, Jonas: How to Train Dependency Parsers with Inexact Search for Joint Sentence Boundary Detection and Parsing of Entire Documents. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, S. 1924–1934, August 2016.

[BK14] Björkelund, Anders; Kuhn, Jonas: Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Pa-

pers). Association for Computational Linguistics, Baltimore, Maryland, USA, S. 47–57, June 2014.

[BN15] Björkelund, Anders; Nivre, Joakim: Non-Deterministic Oracles for Unrestricted Non-Projective Transition-Based Dependency Parsing. In: Proceedings of the 14th International Conference on Parsing Technologies. Association for Computational Linguistics, Bilbao, Spain, S. 76–86, July 2015.

[Co02] Collins, Michael: Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, S. 1–8, July 2002.

[CR04] Collins, Michael; Roark, Brian: Incremental Parsing with the Perceptron Algorithm. In: Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume. Barcelona, Spain, S. 111–118, July 2004.

[DIM05] Daumé III, Hal; Marcu, Daniel: Learning as search optimization: approximate large margin methods for structured prediction. In: ICML. S. 169–176, 2005.

[FdSM12] Fernandes, Eraldo; dos Santos, Cícero; Milidiú, Ruy: Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution. In: Joint Conference on EMNLP and CoNLL - Shared Task. Association for Computational Linguistics, Jeju Island, Korea, S. 41–48, July 2012.

[HFG12] Huang, Liang; Fayong, Suphan; Guo, Yang: Structured Perceptron with Inexact Search. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Montréal, Quebec, Canada, S. 142–151, June 2012.

[Ni08] Nivre, Joakim: Algorithms for Deterministic Incremental Dependency Parsing. Computational Linguistics, 34(4):513–553, 2008.

[Ni09] Nivre, Joakim: Non-Projective Dependency Parsing in Expected Linear Time. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Association for Computational Linguistics, Suntec, Singapore, S. 351–359, August 2009.

[Ro58] Rosenblatt, Frank: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 65(6):386–408, 1958.

[Sm11] Smith, Noah A.: Linguistic Structure Prediction. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, May 2011.

Anders Björkelund, geboren 1985, hat in Lund, Schweden Informatik studiert und einen Master-Abschluss erworben. In Stuttgart promovierte er im Rahmen des Sonderforschungsbereichs 732 „Inkrementelle Spezifikation im Kontext“ am Institut für Maschinelle Sprachverarbeitung mit der Dissertation Online Learning of Latent Linguistic Structure with Approximate Search. Seit 2019 ist er zurück in Schweden und arbeitet am Institut für Astrophysik und Theoretische Physik der Universität Lund.