Entwurf 5

Bachelorarbeit Proposal

Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache

Clemens Damke

Matrikelnr. 7011488

22. Juli 2017

betreut von

Prof. Dr. Eyke Hüllermeier
Intelligente Systeme
Institut für Informatik
Universität Paderborn

1 Motivation und Hintergrund

In den letzten Jahren hat die Repräsentation von Wissensbasen durch Graphen, sog. Wissensgraphen, immer mehr an Bedeutung gewonnen. Google, Bing und IBM Watson benutzen solche Wissensgraphen z. B. zum Beantworten von komplexen Suchanfragen.

Die Grundidee dabei ist es, Entitäten durch Knoten und Relationen durch Kanten abzubilden. Entitäten können konkrete Dinge, wie z. B. Personen, aber auch abstrakte Konzepte, wie z. B. historische Epochen, sein. Relationen beschreiben beliebige Beziehungen zwischen den Entitäten, z. B. $person(\text{Da Vinci}) \xrightarrow{\text{lived in}} epoch(\text{Renaissance})$. Die Entität, von der eine solche Relation ausgeht, wird als Subjekt und die Zielentität als Objekt der Relation bezeichnet.

Da solche Graphen in zahlreichen Domänen einsetzbar sind, wird deren automatisierte Konstruktion bereits seit Jahren erforscht. Manuelles Konstruieren und vor allem anschließendes Warten und Aktualisieren von Wissensgraphen, ist aufgrund der abzubildenden Datenmengen nicht praktikabel. Bei einer maschinellen automatisierten Konstruktion sind insbesondere zwei Anforderungen problematisch:

1. Das Verarbeiten von unstrukturierten Eingaben, wie z. B. natürlichsprachlichen Texten.
2. Effizientes Eingliedern neuer Informationen in einen bestehenden Wissensgraphen. Dieses Eingliedern von Informationen umfasst im Speziellen:
   - **Entity Resolution:** Hinzukommende Entitäten, die bereits im Graphen enthalten sind, müssen als Duplikate erkannt werden. Dies ist i. d. R. nicht trivial, da die selbe Entität durch viele verschiedene, oftmals vom Kontext abhängige, Token repräsentiert werden kann; z. B. Bob vs. Robert oder Der Papst vs. Franziskus.
   - **Link Prediction:** Hinzukommende Entitäten müssen mit bereits bestehenden Entitäten in Relation gesetzt werden. Hinzukommende Relationen können zudem benutzt werden um andere Relationen zu inferieren; z. B.

$$female(A) \wedge B \xrightarrow{\text{son of}} A \implies A \xrightarrow{\text{mother of}} B$$

Die Kombination dieser beiden Anforderungen ist interessant, da das meiste verfügbare Wissen in natürlichsprachlicher Textform vorliegt und zudem permanent neues Wissen entsteht. Ein automatisiertes Wissensgraphkonstruktionsverfahren sollte daher beide Anforderungen berücksichtigen.

Neben diesen Anforderungen bzgl. der Extraktion von Wissen ist zudem wichtig, wie genau der Graph repräsentiert wird. Zusätzlich zu Knoten bzw. Entitäten und Kanten bzw. Relationen sind oftmals weitere Metadaten relevant. Dazu zählt insbesondere die Inferernzkonfidenz des Link Predictors. Da natürlichsprachliche Eingabeinformationen häufig unvollständig oder fehlerhaft sind, ist es für die Interpretation und Analyse des resultierenden Graphen hilfreich jeder Relation eine Konfidenz $\in [0, 1]$ zuzuordnen. Das Ergebnis ist ein sog. probabilistischer Wissensgraph.

2 Ziele der Arbeit

Das beschriebene Problem der online Wissensgraphkonstruktion aus natürlicher Sprache soll im Kontext von textueller Kommunikation zwischen Menschen näher untersucht werden. Gegeben sei ein Stream von Textnachrichten (z. B. E-Mails), denen jeweils ein Inhalt, ein Absender, eine Menge von Empfängern und ggf. weitere Metadaten, wie z. B. Absendezeit, Absendeort oder IP-Adresse, zugeordnet ist. Die Nachrichteninhalte werden der Einfachheit halber als ausschließlich englischsprachig angenommen. Außerdem wird eine, für E-Mails und andere Kurznachrichten typische, eingeschränkte Sprachkomplexität angenommen. Ziel ist es, ein skalierbares Verfahren zu entwickeln, welches aus diesem Nachrichtenstrom einen Wissensgraphen konstruiert. Insbesondere folgende Informationen sollen im resultierenden Graphen abgebildet werden:

1. Jede Nachricht ist eine Entität und soll mit allen zugehörigen Attributen als Knoten eingefügt werden.
2. Jede Person, die eine Nachricht sendet, empfängt oder darin vorkommt, soll als Entität erkannt werden. Personen-Entitäten, die durch eine Entity Resolution mit einer gewissen Konfidenz als identisch erkannt werden, sollen durch eine entsprechend konfidente Äquivalenzkante in Relation gesetzt werden.
3. Neben Personen sollen auch Orte, die in Nachrichten vorkommen, erkannt werden. Idealerweise werden erkannten Orten zudem Geokoordinaten zugeordnet.
4. Jeder Kante soll, sofern sinnvoll und möglich, ein Zeitpunkt oder Zeitraum zugeordnet werden. So kann z. B. die Anwesenheit einer Person an einem Ort temporal einsortiert werden.

Das gesuchte Verfahren soll zudem erweiterbar sein. Es sollen Schnittstellen eingeplant werden, um neben dem Textextraktor auch andere Extraktoren, z. B. für Bilder und Audioaufnahmen, hinzufügen zu können. Außerdem sollen die Entity Resolution und Link Prediction Verfahren um domänenspezifische Expertensysteme erweiterbar sein. Ein Beispiel hierfür ist die Geokoordinatenzuordnung zu extrahierten Ortsnamen. Durch das Erweitern der Link Prediction, kann diese Zuordnung mithilfe eines entsprechenden Expertensystems erfolgen.

Wie zuvor erwähnt, soll das gesuchte Verfahren skalierbar sein. Dies bedeutet konkret, dass die Laufzeit für das Verarbeiten einer Nachricht idealerweise ausschließlich von der Größe der Nachricht und nicht von der bisherigen Größe des Wissensgraphen abhängt. Sofern dies nicht möglich ist, soll evaluiert werden, wie sich der Einfluss der Graphgröße auf die Laufzeit soweit wie möglich reduziert werden. Die Skalierbarkeit des Verfahrens umfasst zudem auch die Eignung für ein massivparalleles Ausführen im Cluster.

Im Rahmen der Arbeit soll ein derartiges Verfahren zuerst entworfen und anschließend prototypisch implementiert werden. Die Skalierbarkeitsanforderungen sollen hierbei primär im

Entwurf des Verfahrens berücksichtigt werden. Es ist ausreichend, wenn das gefundene Verfahren prinzipiell skalierbar (insb. auf mehrere Rechner verteilbar) implementierbar ist, eine solche Implementation muss allerdings nicht erfolgen.

3 Verwandte Arbeiten

3.1 Ontologie

Um einen Wissensgraphen konstruieren zu können, muss zuerst definiert werden, wie genau Konzepte und ihre Beziehungen im Graphen repräsentiert werden. Die formale Definition dieses Repräsentationsschemas ist eine sog. Ontologie. Da das zu konstruierende System keine klar definierte Domäne hat, ist es notwendig ein Gleichgewicht zwischen einer allgemeingültigen und einer praktisch verwendbaren Ontologie zu finden. Eine potentiell geeignete Grundlage für Konstruktion einer solchen Ontologie sind sog. Konzeptgraphen [1, S. 213 ff.]. Konzeptgraphen sind ein graphbasiertes Logikkalkül mit der Mächtigkeit einer Prädikatenlogik erster Ordnung. Sie lassen sich daher zur Beschreibung diverser Domänen einsetzen. Zudem existieren Ersetzungsaxiome um logische Inferenzen über Konzeptgraphen durchzuführen.

Um mit Konzeptgraphen neben formalen Systemen auch natürlichsprachlich kodiertes Wissen zu beschreiben, wurden sie durch das Interoperable Knowledge Representation for Intelligence Support (IKRIS) Projekt um das modallogische Konzept der Proposition erweitert. Die IKRIS Knowledge Language (IKL) [2] spezifiziert diese Erweiterung formal. Eine Ontologie, die auf Konzeptgraphen und IKL aufbaut, ist daher potentiell gut als Wissensgraphschema geeignet. Die Details der zu verwendenden Ontologie müssen im Rahmen dieser Arbeit allerdings noch spezifiziert werden.

3.2 Natural Language Processing

Um den natürlichsprachlichen Inhalt einer Nachricht in den zu konstruierenden Wissensgraphen einfügen zu können, muss dieser zuerst zerlegt und in das Schema der angestrebten Ontologie übersetzt werden. Diese Zerlegung ist ein Teilgebiet des Natural Language Processings (NLP), insbesondere ist hiermit das Dependency Parsing (DP) und die Coreference Resolution (CR) gemeint. Ersteres ermittelt einen sog. Abhängigkeitsgraphen, der die grammatikalische Struktur der Eingabe beschreibt, letztere bestimmt Äquivalenzklassen von Worten, die auf das selbe Konzept verweisen (insbesondere z. B. von Pronomina und ihrem Antezedens). Da Abhängigkeitsgraphen die semantische Struktur eines Textes nur indirekt mittels seiner grammatikalischen Struktur beschreiben, ist neben dem DP und der CR noch eine Transformation in eine zur Ontologie kompatiblen Struktur erforderlich. Die Funktionsweise einer solchen Transformation hängt von der angestrebten Ontologie ab und muss daher im Rahmen dieser Arbeit spezifiziert werden.

Gute geeignete NLP Systeme für den DP und CR Schritt existieren bereits. Populär ist z. B. Stanfords CoreNLP Projekt [3], welches u. a. Module zum Dependency Parsing, zum Erkennen

von Koreferenzen und zum Kategorisieren von Entitäten (Person, Ort, Zeitpunkt, etc.) beinhaltet. Im Rahmen der Arbeit ist zu evaluieren, welche Kombination von Verfahren sich gut als Ausgangspunkt für die Wissensgraphkonstruktion eignet.

3.3 Wissensgraphkonstruktion

Das Einfügen von Nachrichten, die zuvor via NLP in lokale kleine Wissensgraphen übersetzt wurden, in einen bestehenden großen Wissensgraphen und das Schlussfolgern über diesem lässt sich als Teilgebiet des Statistical Relational Learnings (SRL) beschreiben. In der Literatur finden sich im Wesentlichen drei Klassen von Verfahren [4], die auf verschiedenen Annahmen über die Korrelation der zu verarbeitenden Informationen basieren:

1. **Latent Feature Models:** Alle Relationen werden als bedingt unabhängig angenommen, sofern bestimmte Eigenschaften über Subjekte und Objekte der Relationen gegeben sind.
2. **Graph Feature Models:** Alle Relationen werden als bedingt unabhängig angenommen, sofern bestimmte Eigenschaften der Struktur des Graphen gegeben sind.
3. **Markov Random Fields:** Es wird angenommen und erlaubt, dass alle Relationen lokale Abhängigkeiten voneinander haben können.

Ein Beispiel für ein Latent Feature Modell ist das RESCAL-Verfahren [5], welches auf Tensorfaktorisierung basiert. Die Grundidee dabei ist es, allen Entitäten i einen Feature-Vektor $e_i \in \mathbb{R}^H$ zuzuordnen und für alle Relationen k eine Gewichtsmatrix $W_k \in \mathbb{R}^{H \times H}$ zu finden. Die Konfidenz in die Existenz einer Relation $i \xrightarrow{k} j$ wird durch $e_i^T \; W_k \; e_j$ beschrieben. Diese Definition ermöglicht eine sehr schnelle Link Prediction, da lediglich ein Vektor-Matrix-Vektor-Produkt berechnet werden muss. RESCAL liefert gute Ergebnisse, wenn die vorherzusagenden Relationen globale Abhängigkeiten aufweisen. Lokal stark zusammenhängende Teilgraphen werden allerdings schlecht erkannt, da nur der Feature-Vektor und nicht die Nachbarschaft einer Entität berücksichtigt wird; ein Beispiel hierfür sind symmetrische Relationen.

$$A \xrightarrow{\text{married to}} B \implies B \xrightarrow{\text{married to}} A$$

Komplementär zu den Latent Feature Modellen sind die Graph Feature Modelle. Statt Entitäten in einen Feature-Raum einzubetten, wird hier die Nachbarschaft der Entitäten betrachtet. Ein Beispiel hierfür ist der Path Ranking Algorithmus (PRA) [6]. PRA ermittelt Relationen durch zufälliges Durchwandern des Graphen. Um die Stärken der Latent Feature und Graph Feature Modelle zu kombinieren, wurden Hybrid-Modelle, wie z. B. das Additive Relational Effects (ARE) Verfahren [7], entwickelt, welches die Konfidenzen von RESCAL und PRA addiert.

Fundamental verschieden von diesen beiden Verfahren sind Markov Random Fields (MRFs). Hier sind prinzipiell Abhängigkeiten zwischen allen Relationen möglich, was MRFs sehr flexibel macht. Da dies hinsichtlich der Laufzeit schnell impraktikabel wird, wird das Modell um einen Abhängigkeitsgraphen erweitert, der die Anzahl von betrachteten Abhängigkeiten reduziert. Der Abhängigkeitsgraph darf dabei nicht mit dem Wissensgraphen verwechselt werden: Ersterer beschreibt statistische Abhängigkeiten zwischen Klassen von Relationen, während letzterer Abhängigkeiten zwischen spezifischen Fakten beschreibt. Zur Modellierung von

Abhängigkeitsgraphen werden i. d. R. Kalküle verwendet, die an eine Prädikatenlogik erster Ordnung angelehnt sind. Das Finden eines Wissensgraphen ist in diesem Modell äquivalent zum Lösen des MAX-SAT-Problems. Wählt man ein Kalkül in dem die Atome stetig, also aus $[0, 1]$ statt aus $\{0, 1\}$ sind, und Formeln zudem ausschließlich Disjunktionen und Negationen gemäß Łukasiewicz S-Norm enthalten, erhält man ein sog. Hinge-Loss-MRF [8][9], da die Loss-Funktion der Disjunktion in dieser S-Norm ein Hinge-Loss ist. Ein konkretes Kalkül, welches sich zur Spezifikation von HL-MRFs eignet, ist die Probabilistic Soft Logic (PSL) [10][9]. MAX-SAT lässt sich für solche HL-MRFs effizient und parallelisierbar mit dem konvexen Alternating Direction Method of Multipliers (ADMM) Optimierungsverfahren [11] lösen. In dessen ursprünglichen Form ist ADMM allerdings ausschließlich für offline Inferenz geeignet; der Wissensgraph müsste also bei jeder Eingabe neu konstruiert werden. Um dieses Problem zu lösen, wurde das Budgeted Online Collective Inference (BOCI) Verfahren [12] entwickelt. BOCI nutzt Metadaten, die während der Ausführung von ADMM anfallen, um eine Bewertung für jedes Atom zu berechnen. Die Bewertung eines Atoms beschreibt, wie groß die erwartete Wertveränderung beim Eintreffen neuer Informationen ist. Kommen nun neue Informationen an, müssen diese ausschließlich zusammen mit den m höchstbewerteten bereits existierenden Atomen betrachtet werden, die Werte aller anderen Atome werden fixiert. Je höher das Budget m, desto höher ist die Qualität im Vergleich zu einer Neukonstruktion des Graphen. Es wurde empirisch gezeigt, dass die Inferenzqualität mit BOCI oft nur unwesentlich schlechter ist als bei einer kompletten offline Inferenz.

Mittels ADMM und BOCI lässt sich PSL daher für die online Wissensgraphkonstruktion einsetzen. Der Vorteil dieses Ansatzes gegenüber eines Latent Feature oder Graph Feature Modells ist, dass sich andere domänenspezifische Expertensysteme leicht in eine PSL Inferenz integrieren lassen. PSL erlaubt nämlich die Inklusion von benutzerdefinierten Funktionen und Prädikaten. Diese können benutzt werden, um z. B. die Levenshtein-Distanz zweier Zeichenketten mit in die Entity Resolution einfließen zu lassen. Aufgrund des Ziels der Erweiterbarkeit wird PSL daher als Grundlage für diese Arbeit gewählt.

4 Vorläufige Struktur

4.1 Struktur des Verfahrens

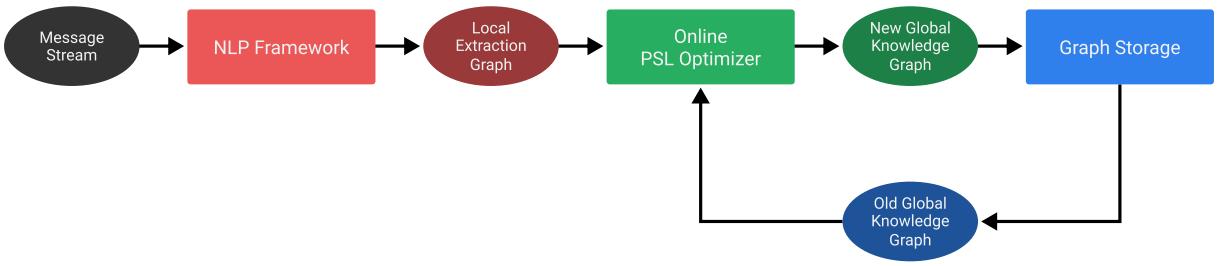

Abbildung 1: Vorläufiges grobes Strukturdiagramm des Verfahrens

Der vorgestellte Ansatz zur Wissensgraphkonstruktion lässt sich in zwei Hauptschritte unterteilen. Diese beiden Schritte werden beim Eintreffen einer Nachricht sequentiell ausgeführt. Im ersten Schritt wird mittels eines NLP-Systems ein lokaler Konzeptgraph extrahiert, der strukturiert den Inhalt der eingetroffenen Nachricht repräsentiert. Hierfür kann z. B. das Stanford CoreNLP Framework [13] benutzt werden.

Im zweiten Schritt wird dieser lokale Konzeptgraph dann in den bereits existierenden Wissensgraphen eingefügt. Mittels eines PSL-Programms werden dann Beziehungen zwischen dem eingefügten Graphen und dem bestehenden Graphen ermittelt. Zur Implementation dieses PSL Programms kann eine bereits existierende PSL Referenzimplementation [14] benutzt werden; diese unterstützt zwar keine Inferenz im Cluster und ausschließlich H2 zur Persistierung, für eine prototypische Implementation ist dies allerdings ausreichend.

4.2 Inhaltsverzeichnis

1. Einleitung
    - Hintergrund
    - Problemstellung
    - Ziele
2. Literaturüberblick
    - Ansätze zur Wissensrepräsentation
    - Generelle Konstruktionsansätze für Wissensbasen
    - NLP Verfahren
    - Ansätze zur automatisierten Wissensgraphkonstruktion
3. Theoretische Grundlagen
    - Modellierung von Wissen mittels Konzeptgraphen
    - Dependency Parsing und Coreference Resolution
    - Modellierung von Hinge-Loss-MRFs mit PSL
4. Vorgeschlagenes Wissensgraphkonstruktionsverfahren
    - Grundidee
    - Wissensgraphontologie
    - Graph-Persistenzschicht
    - NLP-Phase
    - Graphkonstruktionsphase
5. Auswertung
    - Testmethode

- Ergebnisse
- Zusammenfassung und Ausblick

4.3 Zeitplan

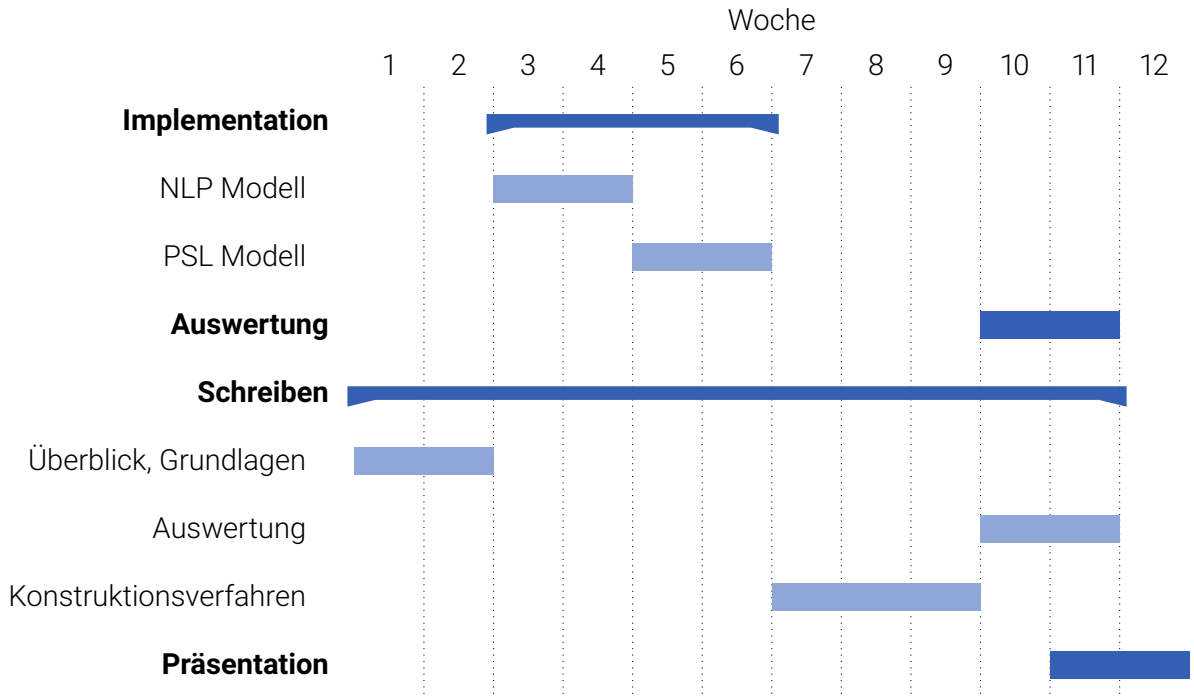

Abbildung 2: Vorläufiger Zeitplan

Literaturverzeichnis

[1] Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. Handbook of Knowledge Representation. Elsevier Science, San Diego, USA, 2007. ISBN 0444522115, 9780444522115. URL `http://dai.fmph.uniba.sk/~sefranek/kri/handbook/handbook_of_kr.pdf`.

[2] Pat Hayes. The IKRIS Knowledge Language. URL `https://www.ihmc.us/users/phayes/IKL/GUIDE/GUIDE.html`. Zuletzt besucht im Mai 2017.

[3] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In Association for Computational Linguistics (ACL) System Demonstrations, pages 55–60, 2014. URL `http://www.aclweb.org/anthology/P/P14/P14-5010.pdf`.

[4] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. In Proceedings of the IEEE, volume 104. Institute of Electrical and Electronics Engineers (IEEE), 2016. doi: 10.1109/jproc.2015.2483592. URL `https://arxiv.org/pdf/1503.00759.pdf`.

[5] Maximilian Nickel. Tensor Factorization for Relational Learning. PhD thesis, Ludwig-Maximilians-Universität München, 2013. URL https://edoc.ub.uni-muenchen.de/16056/1/Nickel_Maximilian.pdf.

[6] Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pages 529–539, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL https://www.cs.cmu.edu/~tom/pubs/lao-emnlp11.pdf.

[7] Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Reducing the rank in relational factorization models by including observable patterns. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 1179–1187. Curran Associates, Inc., 2014. URL http://www.dbs.ifi.lmu.de/~tresp/papers/nips2014_final.pdf.

[8] Stephen H. Bach, Bert Huang, Ben London, and Lise Getoor. Hinge-loss markov random fields: Convex inference for structured prediction. In Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI'13, pages 32–41, Arlington, Virginia, United States, 2013. AUAI Press. URL https://arxiv.org/ftp/arxiv/papers/1309/1309.6813.pdf.

[9] Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss markov random fields and probabilistic soft logic. ArXiv:1505.04406 [cs.LG], 2015. URL https://linqspub.soe.ucsc.edu/basilic/web/Publications/2015/bach:arxiv15/bach-arxiv15.pdf.

[10] Matthias Bröcheler, Lilyana Mihalkova, and Lise Getoor. Probabilistic similarity logic. In Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI'10, pages 73–82, Arlington, Virginia, United States, 2010. AUAI Press. ISBN 978-0-9749039-6-5. URL https://event.cwi.nl/uai2010/papers/UAI2010_0089.pdf.

[11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn., 3(1):1–122, January 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL http://stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf.

[12] Jay Pujara, Ben London, and Lise Getoor. Budgeted online collective inference. In Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI'15, pages 712–721, Arlington, Virginia, United States, 2015. AUAI Press. ISBN 978-0-9966431-0-8. URL http://psl.linqs.org/files/pujara-uai15.pdf.

[13] Stanford CoreNLP. URL https://stanfordnlp.github.io/CoreNLP. Zuletzt besucht im Juli 2017.

[14] PSL Referenzimplementation. URL https://github.com/linqs/psl. Zuletzt besucht im Juli 2017.