

# Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache

---

Clemens Damke

*31. August 2017*

Version: Entwurf 1





**PADERBORN UNIVERSITY**  
*The University for the Information Society*

Department of Electrical Engineering,  
Computer Science and Mathematics  
Warburger Straße 100  
33098 Paderborn



Intelligent Systems Group (ISG)

Bachelorarbeit

# **Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache**

Clemens Damke

- |                     |  |
|---------------------|--|
| <i>1. Korrektor</i> | <b>Prof. Dr. Eyke Hüllermeier</b><br>Institut für Informatik<br>Universität Paderborn          |
| <i>2. Korrektor</i> | <b>Prof. Dr. Axel-Cyrille Ngonga Ngomo</b><br>Institut für Informatik<br>Universität Paderborn |
| <i>Betreuer</i>     | Dr. Theodor Lettmann und Prof. Dr. Eyke Hüllermeier  |

31. August 2017

**Clemens Damke**

*Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache*

Bachelorarbeit, 31. August 2017

Korrektoren: Prof. Dr. Eyke Hüllermeier und Prof. Dr. Axel-Cyrille Ngonga Ngomo

Betreuer: Dr. Theodor Lettmann und Prof. Dr. Eyke Hüllermeier

**Universität Paderborn**

*Intelligente Systeme*

Institut für Informatik

Pohlweg 51

33098 Paderborn





# Abstract

Hallo Welt. Test5.





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Ziele der Arbeit . . . . .	4
1.3	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Verwandte Arbeiten</b>	<b>7</b>
2.1	Ansätze zur Wissensrepräsentation . . . . .	7
2.1.1	Logische Grundlagen . . . . .	7
2.1.2	Entwicklung maschineller Wissensrepräsentation . . . . .	9
2.1.3	Aktuelle Wissensrepräsentationsansätze . . . . .	10
2.2	Konstruktionsansätze für Wissensgraphen . . . . .	11
2.3	NLP Werkzeuge . . . . .	11
<b>3</b>	<b>Theoretische Grundlagen</b>	<b>13</b>
3.1	Wissensmodellierung mit Konzeptgraphen . . . . .	13
3.2	Dependency Parsing und Coreference Resolution . . . . .	13
3.3	Modellierung von Hinge-Loss-MRFs mit PSL . . . . .	13
<b>4</b>	<b>Vorgeschlagenes Wissensgraphkonstruktionsverfahren</b>	<b>15</b>
4.1	Wissensgraphontologie . . . . .	15
4.2	Graph-Persistenzschicht . . . . .	15
4.3	NLP-Phase . . . . .	15
4.4	Graphkonstruktionsphase . . . . .	15
<b>5</b>	<b>Auswertung</b>	<b>17</b>
5.1	Testmethode . . . . .	17
5.2	Ergebnisse . . . . .	17
<b>6</b>	<b>Zusammenfassung</b>	<b>19</b>
<b>A</b>	<b>Anhang</b>	<b>23</b>







# Einleitung

“ *The actual world cannot be distinguished from a world of imagination by any description. Hence the need of pronoun and indices, and the more complicated the subject the greater the need of them.*

— **Charles Sanders Peirce**  
Mathematiker und Philosoph

## 1.1 Motivation

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.

Die Typen von Entitäten bzw. Relationen (z. B. *person* bzw. *lived in*) und deren Bedeutung sind dabei i. d. R. formal in einer sog. Ontologie spezifiziert. Die Ontologie beschränkt also die Menge gültiger Wissensgraphen, was eine effiziente maschinelle Verarbeitung der im Graph enthaltenen Informationen ermöglicht.

Da Wissensgraphen 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, welches beide Anforderungen berücksichtigt, ist daher in diversen Domänen von Nutzen. Ein Beispiel hierfür ist die Auswertung von Kommunikationsdaten aus E-Mails oder Chat-Nachrichten mit dem Ziel die sozialen Beziehungen und Intentionen der Kommunikationspartner zu ermitteln.

## 1.2 Ziele der Arbeit

Das übergeordnete Ziel dieser Arbeit ist es, ein Verfahren zu finden, welches das soeben beschriebene Problem der automatisierten Wissensgraphkonstruktion für E-Mail-Daten löst. Konkret sei ein Stream von E-Mails gegeben, 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. Die Nachrichten sollen nacheinander in das zu konstruierende System eingefügt werden, welches sukzessive einen Wissensgraphen daraus erzeugt.

Für diese Erzeugung muss eine Reihe von Teilproblemen gelöst werden:

1. **Onotologie:** Spezifikation einer Wissensgraphontologie, die mächtig genug ist, um die Diversität natürlichsprachlich beschriebener Informationen abzubilden.
2. **Repräsentation:** Spezifikation der maschinellen Repräsentation des Wissensgraphen.
3. **Sprachverarbeitung:** Finden eines Verfahrens, welches die natürlichsprachlichen Inhalte der Nachrichten in eine für die Wissensgraphkonstruktion geeignete Form bringt.
4. **Grapherweiterung:** Finden eines Verfahrens, um eine eintreffende Nachricht in den bestehenden Wissensgraphen einzufügen.

Das aus den Teillösungen zusammengesetzte Verfahren muss, neben der offensichtlichen Anforderung einen Wissensgraphen zu konstruieren, zudem folgende technische Anforderungen erfüllen:

1. **Erweiterbarkeit:** Es sollen Schnittstellen eingeplant sein, um neben der Sprachverarbeitung auch andere Verarbeitungsverfahren, z. B. für Bilder, hinzufügen zu können. Die Graphontologie, Graphrepräsentation und das Grapherweiterungsverfahren dürfen also nicht zu sehr auf die Struktur natürlicher Sprache zugeschnitten sein.
2. **Parallelisierbarkeit:** Das Verfahren soll in der Lage sein die Rechenleistung mehrerer Prozessorkerne zu nutzen. Diese Anforderung betrifft insbesondere das Grapherweiterungsverfahren.

## 1.3 Aufbau der Arbeit

Kapitel 2

Kapitel 3

Kapitel 4

Kapitel 5

## Kapitel 6



## Verwandte Arbeiten

Die in 1.2 beschriebenen Ziele werden bereits seit langem erforscht. Der Begriff *Wissensgraph* wurde 2012 durch Google popularisiert, die Ideen dahinter lassen sich allerdings bis ins Ende des 19. Jahrhunderts zurückverfolgen. Dieses Kapitel zeigt auf, wie sich die Themen dieser Arbeit in die bisherige Forschung einfügen. 2.1 ordnet das Konzept des Wissensgraphen in die Entwicklungsgeschichte der Wissensrepräsentation ein. 2.2 beschreibt die aktuell verwendeten Verfahren zur Konstruktion von Wissensgraphen. In 2.3 wird schließlich ein Überblick über die momentan verbreiteten NLP (*natural language processing*) Werkzeuge gegeben.

## 2.1 Ansätze zur Wissensrepräsentation

### 2.1.1 Logische Grundlagen

Die Entwicklung der Wissensrepräsentation hängt eng mit der Entwicklung der Logik zusammen. Während in der formalen Logik und Mathematik die Prädikatenlogik das allgemein verwendete Kalkül ist und alternative Formalismen kaum verbreitet sind, finden im Bereich der Wissensrepräsentation bis heute diverse andere Kalküle Verwendung. Diese werden im Folgenden kurz vorgestellt.

**Begriffsschrift (1879)** Gottlob Freges Buch über die *Begriffsschrift* gilt als eines der bedeutsamsten Werke der Logik. Sie ist der erste Formalismus mit der Mächtigkeit der modernen Prädikatenlogik zweiter Ordnung mit Identität. Frege benutzt hierfür eine zweidimensionale Notation, die sich stark von der heute gebräuchlichen, linearen, an die Algebra angelehnte Notation unterscheidet.

$$\begin{array}{c} \vdash^a \quad \vdash \\ \vdash \quad \vdash \end{array} \begin{array}{c} \mathfrak{A}(a) \\ \mathfrak{B}(a) \end{array} \quad \Leftrightarrow \quad \exists a : P(a) \vee R(a)$$

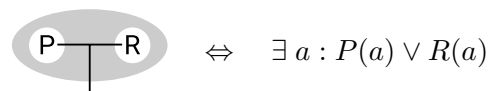
Im Gegensatz zur Prädikatenlogik gibt es keine eigene Syntax für *UND* und *ODER*; diese Operatoren müssen durch die Kombination von Negation und Implikation abgebildet werden. Zudem gibt es ausschließlich den Allquantor; eine existenzquantisierte Aussage muss durch Negation der negierten allquantisierten Aussage ausgedrückt werden.

**Prädikatenlogik: Peirce Notation (1885)** Unabhängig von Frege entwickelte der amerikanische Mathematiker Charles Sanders Peirce ebenfalls ein prädikatenlogisches Kalkül. Peirces Notation hatte starke Ähnlichkeiten mit der heute benutzten linearen Schreibweise. Statt den modernen Symbolen hat Peirce allerdings die algebraischen Operatoren benutzt, um die Analogien zwischen Logik und Algebra auszudrücken.

$$\Sigma_a P_a + R_a \Leftrightarrow \exists a : P(a) \vee R(a)$$

**Existential Graphs (1897)** Neben seiner zuvor entwickelten linearen prädikatenlogischen Notation, hat Peirce zudem viele Jahre an einem alternativen graphischen Kalkül gearbeitet, welches er *Existential Graphs* (zu dt. Existenzgraphen) nannte. Ähnlich wie die Begriffsschrift werden Existenzgraphen zweidimensional dargestellt. Von dieser Gemeinsamkeit abgesehen, funktionieren sie allerdings fundamental verschieden. Ein logischer Ausdruck wird hier durch einen ungerichteten Graphen beschrieben. Die konkrete räumliche Anordnung der Knoten und Kanten hat dabei keine semantische Relevanz.

Peirce hat Existenzgraphen als ein dreistufiges aufeinander aufbauendes System konzipiert. Die erste Stufe, die sog.  $\alpha$ -Graphen, umfasst alle notwendigen syntaktischen Elemente, um ein Kalkül mit der Mächtigkeit der Aussagenlogik zu erhalten. Die  $\beta$ -Graphen bilden die zweite Stufe und erweitern die Syntax der  $\alpha$ -Graphen, sodass die Mächtigkeit der Prädikatenlogik erster Ordnung erreicht wird. Sowohl für  $\alpha$ -, als auch für  $\beta$ -Graphen, ist die Vollständigkeit und Korrektheit bewiesen. Die dritte Stufe ( $\gamma$ -Graphen) wurde von Pierce nie vollendet; sie deckt in etwa die Mächtigkeit der heutigen Prädikatenlogik höherer Ordnung sowie der Modallogik ab.



Wie schon die Begriffsschrift, sind Existenzgraphen syntaktisch minimal. Direkt ausdrücken lässt sich lediglich *UND*, der Existenzquantor und die Negation. Ein weiterer Unterschied zur heutigen Prädikatenlogik ist die Beschreibung logischer Inferenzen. Im Gegensatz zu den prädikatenlogischen Ersetzungsaxiomen, die auf der syntaktischen Struktur von logischen Ausdrücken operieren (z. B. für Kommutativität), lassen sich die Ersetzungsaxiome für Existenzgraphen als Graphtransmutationsregeln verstehen, die bestimmte Teilmengen der Knoten und Kanten eines Ausdrucks durch andere äquivalente Knoten- und Kantenmengen ersetzen.

**Prädikatenlogik: Peano-Russell Notation (1910)** Die zweidimensionalen Notationen wurde häufig kritisiert, da sie die lineare, algebraische Notation der symbolischen Logik von Boole und De Morgan verwarfen. Freges Begriffsschrift und Peirces

Existenzgraphen konnten sich daher nicht durchsetzen. Peirces algebraische prädikatenlogische Notation hingegen, stieß auf größere Akzeptanz. Giuseppe Peano hat auf deren Basis eine ähnliche Notation entwickelt, welche allerdings nicht die algebraischen Operatoren benutzt, damit sich logische Ausdrücke besser mit mathematischen Ausdrücken kombinieren lassen. Bertrand Russell hat Peanos Notation anschließend in leicht abgewandelter Form in den *Principia Mathematica* (1910) benutzt. Diese sog. Peano-Russell-Notation ist im Wesentlichen identisch mit der modernen Schreibweise.

Trotz des Verschwindens der zweidimensionalen Notationen, finden sich noch heute Anlehnungen daran. So ist z. B. die Negation  $\neg A$  auf Freges negierten Inhaltsstrich  $\neg\text{—}$   $A$  und der Ableitungsoperator  $\vdash$  auf Freges Urteilsstrich mit angefügtem Inhaltsstrich  $\vdash\text{—}$  zurückzuführen.

## 2.1.2 Entwicklung maschineller Wissensrepräsentation

Die Idee Computer zur Lösung beliebiger Probleme zu benutzen ist nicht neu. Da ein solches maschinelles Problemlösen immer die Verfügbarkeit von Hintergrundwissen über die Problemdomäne erfordert, wurden Methoden zur Wissensrepräsentation immer im Zusammenhang mit Problemlösern erforscht. So wie effiziente Datenstrukturen die Implementation effizienter Algorithmen ermöglichen, ermöglichen gute Wissensrepräsentationen die Implementation guter Problemlöser. Was genau nun als ein guter Problemlöser verstanden wird, hat sich im Laufe der Jahre allerdings immer wieder verändert.

**Universelle Problemlöser** Einer der ersten maschinellen Problemlöser war der von Simon, Shaw und Newell 1955 entwickelte *Logic Theorist* (LT). LT war in der Lage logische Aussagen zu beweisen, indem er systematisch Ersetzungsaxiome auf eine gegebene Aussage angewandt hat, bis die gesuchte Lösung abgeleitet wurde.

Die Grundidee des LT haben Simon, Shaw und Newell 1959 im *General Problem Solver* (GPS) erweitert. Es wurden Heuristiken hinzugefügt, um den Suchraum geschickter zu durchlaufen. GPS war ein universeller Problemlöser, konnte also jedes Problem lösen, das sich durch eine Menge von Horn-Klauseln ausdrücken lässt. Zwar war es so theoretisch möglich Probleme aus diversen Domänen zu lösen, aufgrund der kombinatorischen Explosion war GPS allerdings nicht zur Lösung komplexer praktischer Probleme geeignet.

**Expertensysteme** Aufgrund der Misserfolge universeller Problemlöser für praktische Probleme, hat die Forschung begonnen sich mehr auf die Entwicklung von

Expertensystemen zu fokussieren. Expertensysteme besitzen für gewöhnlich eine Wissensbasis in der domänenspezifisches Wissen in Form von Regeln und Fakten kodiert ist. Eine sog. Inferenzmaschine benutzt diese Regeln und Fakten um Probleme zu lösen.

**Semantic Networks (1956)** Die Idee, Graphen als Datenstruktur für Wissensbasen zu verwenden, taucht erstmal in den sog. *Semantic Networks* (zu dt. semantische Netze) auf. Dieser Ansatz beschreibt Wissen als Menge von (*subject, predicate, object*)-Tripeln. Es gibt darüber hinaus keine klaren Regeln, wie ein semantisches Netz strukturiert sein muss.

**Conceptual Graphs (1976)** Wie genau mit Graphen komplexes Wissen beschrieben werden kann, das über eine reine Taxonomie hinaus geht, blieb bei semantischen Netzen unklar. John F. Sowa's *Conceptual Graphs* (zu dt. Konzeptgraphen) lösen dieses Problem. Statt Wissen als eine Menge von Beziehungen abzubilden, wird es als prädikatenlogischer Ausdruck verstanden. Hierfür baut Sowa auf Peirces Existenzgraphen auf, die bis dahin weitestgehend unbeachtet waren.

Dieser Ansatz erlaubt es komplexe Wissensbasen zu konstruieren, in denen nicht nur gespeichert werden kann, ob ein Konzept existiert, sondern auch, ob es nicht oder nur möglicherweise existiert.

**Knowledge Graphs (1992)** Der Begriff *Knowledge Graph* (zu dt. Wissensgraph) bezeichnete ursprünglich eine Klasse semantischer Netze, deren Relationsmenge formal spezifiziert ist. Dies schränkt die Menge erlaubter Graphen ein und ermöglicht die Definition von Inferenzregeln, um Schlussfolgerungen aus einem gegebenen Graphen zu ziehen. Im Laufe der Jahre ist die Grenze zwischen semantischen Netzen und Wissensgraphen allerdings immer weiter verschwommen, sodass die Bezeichnungen heute oft synonym verwendet werden. Wissensgraphen und Konzeptgraphen müssen weiterhin unterschieden werden, da erstere oftmals Negation und Modalität nicht unterstützen.

### 2.1.3 Aktuelle Wissensrepräsentationsansätze

#### Semantic Web

#### NELL

#### Google Knowledge Graph

## 2.2 Konstruktionsansätze für Wissensgraphen

## 2.3 NLP Werkzeuge



## Theoretische Grundlagen

- 3.1 Wissensmodellierung mit Konzeptgraphen
- 3.2 Dependency Parsing und Coreference Resolution
- 3.3 Modellierung von Hinge-Loss-MRFs mit PSL





## Vorgeschlagenes Wissensgraph-konstruktionsverfahren

### 4.1 Wissensgraphontologie

### 4.2 Graph-Persistenzschicht

### 4.3 NLP-Phase

### 4.4 Graphkonstruktionsphase



## Auswertung

### 5.1 Testmethode

### 5.2 Ergebnisse



# Zusammenfassung

6









Anhang

A



# Abbildungsverzeichnis



## Tabellenverzeichnis









# Erklärung zur Bachelorarbeit

Ich, Clemens Damke (Matrikel-Nr. 7011488), versichere, dass ich die Bachelorarbeit mit dem Thema *Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen der Arbeit, die ich anderen Werken dem Wortlaut oder dem Sinn nach entnommen habe, wurden in jedem Fall unter Angabe der Quellen der Entlehnung kenntlich gemacht. Das Gleiche gilt auch für Tabellen, Skizzen, Zeichnungen, bildliche Darstellungen usw. Die Bachelorarbeit habe ich nicht, auch nicht auszugsweise, für eine andere abgeschlossene Prüfung angefertigt. Auf § 63 Abs. 5 HZG wird hingewiesen.

*Paderborn, 31. August 2017*

---

Clemens Damke