

# Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache

---

Clemens Damke

*28. 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  |

28. August 2017

**Clemens Damke**

*Probabilistische online Wissensgraphkonstruktion aus natürlicher Sprache*

Bachelorarbeit, 28. 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
2.2	Konstruktionsansätze für Wissensbasen . . . . .	7
2.3	NLP Verfahren . . . . .	7
<b>3</b>	<b>Theoretische Grundlagen</b>	<b>9</b>
3.1	Wissensmodellierung mit Konzeptgraphen . . . . .	9
3.2	Dependency Parsing und Coreference Resolution . . . . .	9
3.3	Modellierung von Hinge-Loss-MRFs mit PSL . . . . .	9
<b>4</b>	<b>Vorgeschlagenes Wissensgraphkonstruktionsverfahren</b>	<b>11</b>
4.1	Wissensgraphontologie . . . . .	11
4.2	Graph-Persistenzschicht . . . . .	11
4.3	NLP-Phase . . . . .	11
4.4	Graphkonstruktionsphase . . . . .	11
<b>5</b>	<b>Auswertung</b>	<b>13</b>
5.1	Testmethode . . . . .	13
5.2	Ergebnisse . . . . .	13
<b>6</b>	<b>Zusammenfassung</b>	<b>15</b>
<b>A</b>	<b>Anhang</b>	<b>19</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. Spezifikation einer Wissensgraphontologie, die mächtig genug ist, um die Diversität natürlichsprachlich beschriebener Informationen abzubilden.
2. Spezifikation der maschinellen Repräsentation des Wissensgraphen.
3. Finden eines Verfahrens, welches die natürlichsprachlichen Inhalte der Nachrichten in eine maschinell leichter zu verarbeitende Form bringt.
4. Finden eines Verfahrens, um eine eintreffende Nachricht in den bestehenden Wissensgraphen einzufügen.

## 1.3 Aufbau der Arbeit

**Kapitel 2**

**Kapitel 3**

**Kapitel 4**

**Kapitel 5**

**Kapitel 6**





## Verwandte Arbeiten

2.1 Ansätze zur Wissensrepräsentation

2.2 Konstruktionsansätze für Wissensbasen

2.3 NLP Verfahren



## 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, 28. August 2017*

---

Clemens Damke