



Lehrstuhl Angewandte Informatik IV  
Datenbanken und Informationssysteme  
Prof. Dr.-Ing. Stefan Jablonski

Institut für Angewandte Informatik  
Fakultät für Mathematik, Physik und Informatik  
Universität Bayreuth

## Bachelorarbeit

---

Philipp Scholz

*23.April 2019*

Version: Draft



# Universität Bayreuth

Fakultät Mathematik, Physik, Informatik

Institut für Informatik

Lehrstuhl für Angewandte Informatik IV

NLP-Plattform: Integration und Evaluation von POS-Tagging-Algorithmen

## Bachelorarbeit

Philipp Scholz

- |                    |   |
|--------------------|---|
| <i>1. Reviewer</i> | <b>Dr. Lars Ackermann</b><br>Fakultät Mathematik, Physik, Informatik<br>Universität Bayreuth              |
| <i>2. Reviewer</i> | <b>Prof. Dr.-Ing. Stefan Jablonski</b><br>Fakultät Mathematik, Physik, Informatik<br>Universität Bayreuth |
| <i>Supervisors</i> | Lars Ackermann and Stefan Jablonski   |

23.April 2019

**Philipp Scholz**

*Bachelorarbeit*

NLP-Plattform: Integration und Evaluation von POS-Tagging-Algorithmen, 23.April 2019

Reviewers: Dr. Lars Ackermann and Prof. Dr.-Ing. Stefan Jablonski

Supervisors: Lars Ackermann and Stefan Jablonski

**Universität Bayreuth**

*Lehrstuhl für Angewandte Informatik IV*

Institut für Informatik

Fakultät Mathematik, Physik, Informatik

Universitätsstrasse 30

95447 Bayreuth

Germany

# Abstrakt

Im Themenbereich *Natural Language Processing* (kurz NLP) versucht die Informatik, natürliche Sprachen für Algorithmen zugänglich und interpretierbar zu machen. Ein wichtiger Zwischenschritt ist die Identifizierung der syntaktischen Bedeutung von Wörtern (*Parts of Speech*, kurz POS) in gesprochener Sprache, und deren Zuweisung in Form von Tags (POS-Tagging). Für diese Aufgabe existiert eine Vielzahl unterschiedlich leistungsfähiger und robuster Algorithmen.

Im Rahmen dieser Arbeit wurde eine Sammlung solcher POS-Tagging-Algorithmen zusammen mit einem Evaluationssystem als Operatoren im Programm RapidMiner (zur Verfügung gestellt von RapidMiner GmbH) implementiert.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Aufbau der Arbeit . . . . .	1
<b>2</b>	<b>Allgemein</b>	<b>3</b>
2.1	Tagging-Ansätze . . . . .	3
2.2	Probleme und Ziele . . . . .	3
2.2.1	Tokenization-Unterschiede . . . . .	3
<b>3</b>	<b>Implementierung</b>	<b>5</b>
3.1	Planung . . . . .	5
3.1.1	Inputformat . . . . .	5
3.1.2	Format für Tagger-Ergebnisse . . . . .	6
3.2	Struktur . . . . .	6
3.2.1	Übersicht . . . . .	6
3.3	Evaluation . . . . .	8
3.3.1	Parsing . . . . .	8
3.3.2	Berechnete Werte . . . . .	8
3.4	Conclusion . . . . .	8
<b>4</b>	<b>Concepts</b>	<b>9</b>
4.1	Concepts Section 1 . . . . .	9
4.2	Concepts Section 2 . . . . .	9
4.3	Concepts Section 3 . . . . .	9
4.4	Conclusion . . . . .	9
<b>5</b>	<b>Conclusion</b>	<b>11</b>
5.1	System Section 1 . . . . .	11
5.2	System Section 2 . . . . .	12
5.3	Future Work . . . . .	14





# Einleitung

:TODO :NC

Keine bekannte Lebensform hat eine mit der menschlichen Sprache vergleichbar komplexe Form von Informationsaustausch entwickelt [Rao:2018]. Von selbst ergibt sich die Frage, ob und wie Sprache maschinell verarbeitet werden kann, um sie unter anderem zu interpretieren oder zusammenzufassen. Antworten auf diese Problemstellung liefert das Teilgebiet der Informatik *Natural Language Processing* (Dt. Verarbeitung natürlicher Sprachen, kurz NLP). NLP gliedert sich in viele Teilbereiche: Strukturelle Analyse, Semantik, Phonetik und einige weitere. In dieser Arbeit konzentrieren wir uns auf einen wichtigen Bestandteil der strukturellen Analyse, dem korrekten Identifizieren von syntaktischen Rollen von Wörtern (*Parts of Speech*, kurz POS), bezeichnet als *POS-Tagging*.

Ein Algorithmus, der POS-Tagging betreibt (POS-Tagger), nimmt die Sprache in Textform an und gibt ihn üblicherweise mit Tags versehen wieder aus, wie beispielsweise der Text

*I like the blue house.*

zu folgendem verarbeitet wird:

*I\PRONOUN like\VERB the\DET blue\ADJ house\NOUN .\.*

Da POS-Tagger nicht garantiert fehlerfrei arbeiten (siehe Kap. 2), ist es interessant, deren Performance zu bewerten.

## 1.1 Aufbau der Arbeit

In Kapitel 2 wird

Chapter ??

Chapter 4

## Chapter 4

## Chapter 5

# Allgemein

Um einheitliche Verarbeitung und Vergleichbarkeit zu ermöglichen, werden die Tags in *Tagsets* definiert, wie zum Beispiel dem des Penn-Treebank-Projekts [Web:PennBank:2003].

Betrachtet man das Wort „like“ aus dem einleitenden Beispiel „I like the blue House.“, dann fällt auf, dass es alternativ zum Verb „mögen“ auch als Präposition „wie“ interpretiert werden könnte, auch wenn der Satz dann keinen Sinn mehr ergibt. Diese Uneindeutigkeit (*Ambiguität*) ist das zentrale zu lösende Problem für POS-Tagger [Smith:2011]; Im Gegensatz zum Menschen kann ein Algorithmus Ambiguitäten nicht intuitiv auflösen, sondern muss auf statistische, rechnerische und linguistische Methoden sowie Wissen aus diesen Teilbereichen zurückgreifen.

## 2.1 Tagging-Ansätze

## 2.2 Probleme und Ziele

Da Tagging-Algorithmen sowohl nicht fehlerfrei arbeiten als auch in ihrem Ergebnisformat voneinander - und insbesondere von Goldstandards - abweichen, entstehen Unsicherheiten beim Arbeiten mit solchen Informationen. Welche Probleme besonders dominant sind und welche Zielsetzungen für das Projekt sich daraus ergeben, wird in den folgenden Abschnitten ausführlicher behandelt.

### 2.2.1 Tokenization-Unterschiede

Ein wichtiger Schritt zur Vorverarbeitung von Text ist es, ihn in *Tokens* zu zerlegen. Ein Token ist ein Abschnitt, zu dem ein Tag gehört, typischerweise einzelne Wörter oder Satzzeichen. Formal wird aus dem zusammenhängenden Wort beziehungsweise Text  $w$  eine Menge  $M$  von  $n$  Teilwörtern  $c_i$

$$M = \{c_1, c_2, \dots, c_n\}$$

generiert. POS-Tagger wie der von NLP4J :NC und der Stanford University :NC übernehmen diese Zerlegung selbst. Nehmen wir nun an,  $M$  wäre die Zerlegung von  $w$  in einem Goldstandard, und ein Tagger produziert ein Ergebnis mit der Zerlegung:

$$M_{tag} = \{d_1, d_2, \dots, d_m\}$$

Hierbei sind  $d_i$  wieder Teilworte und  $m$  die Anzahl dieser. Spätestens wenn nun  $n \neq m$  gilt, wird klar, dass  $M$  und  $M_{tag}$  nicht mehr einfach iterativ verglichen werden können, da ab der Stelle  $p$ , wo der Text unterschiedlich geteilt wurde,  $c_i \neq d_i$  sein kann, wobei  $(p \leq i \leq \min\{n, m\})$ . Es ist sogar nicht auszuschließen, dass solche Fehler bereits vorher passieren, obwohl weiterhin  $n = m$  gilt.

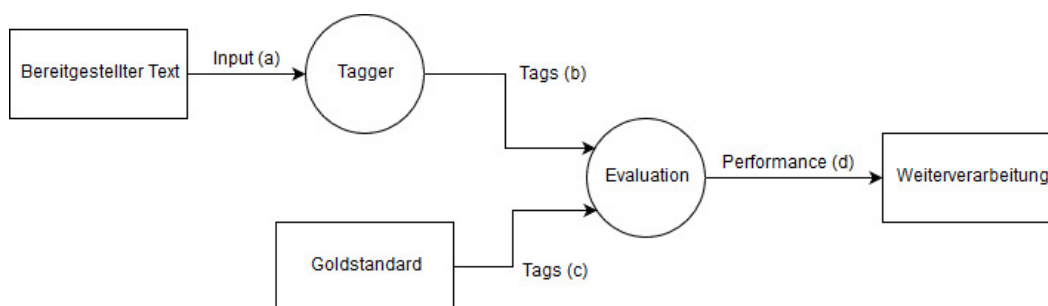
Eines der Ziele bei der Implementierung ist also, den Vergleich von zwei Token-Mengen robust gegen solche Fehler zu machen.

# Implementierung

Dieses Kapitel behandelt die Zielsetzungen, konkrete Planung und die Implementierung der RapidMiner-Erweiterung.

## 3.1 Planung

Zuerst müssen wir betrachten, welche Begebenheiten die RapidMiner-Plattform uns bietet, d.h. welche Datentypen ein Parser annehmen muss, wie man Ergebnisse möglichst ohne Informationsverlust übergibt und Goldstandards richtig einliest.



**Abb. 3.1:** Datenfluss in einem typischen Tagging- und Evaluationsprozess

Eine Übersicht hierzu bietet Abbildung 3.1, wobei insbesondere der Input (a) und Output (b) des Taggers interessant sind. Das Format des Goldstandards (c) ist exogen vorgegeben und das Ergebnis des Evaluationsprozesses muss lediglich RapidMiner-konform kodiert werden.

### 3.1.1 Inputformat

Für die Eingabe an den Tagger wäre es optimal, wenn Outputs anderer Textverarbeitender RapidMiner-Erweiterungen, insbesondere vom *Text Processing Plugin* :NC angenommen werden können. Hierzu kann einfach das Übergabeformat *Document* aus diesem Plugin verwendet werden.

### 3.1.2 Format für Tagger-Ergebnisse

Abhängig vom Tagger können Ergebnisse sehr Informationsreich sein. Der LingPipe-POS-Tagger liefert beispielsweise pro Token (Wort oder Symbol) nicht nur nicht nur das aus seiner Sicht plausibelste Tag (*First-Best*), sondern auch  $n-1$  weitere, nachstehende Optionen (*N-Best*), wobei  $n$  gewählt werden kann :NC . Eine solche Menge an Metainformationen kann der Standard-Datentyp *Document* seitens RapidMiner nur schwer Kodieren. Eine neu entworfene Datenstruktur, die von Operator zu Operator übergeben werden können, bietet neue Freiheiten:

**Segmentierung** der Tag- und Token-Kette anhand von bestimmten Tags, die immer richtig erkannt werden, macht robuster gegen Verschiebungen im Ergebnis wie in Abschnitt 2.2.1 beschrieben: Wird Segment-weise verglichen und entstehen Verschiebungen durch ungleiche Token-Zerlegungen, dann verschwindet dieser Fehler, sobald ein neues Segment betreten wird, da die neuen betrachteten Segmente wieder an gleicher Stelle beginnen.

**Anreicherung** mit Zusatzinformationen ist möglich, da im Gegensatz zu einer einfachen Textuellen Darstellung auf ein Token nicht exakt ein Tag folgen muss. Hier kann beispielsweise pro Tag eine beliebig lange Liste der *N-Best* Tags stehen.

Ein solches alternatives Format ist jedoch von Operatoren, die nicht speziell dafür vorbereitet sind (wie der Evaluationsoperator), nicht lesbar. Darum müssen Ergebnisse auch immer als Typ *Document* angegeben werden.

## 3.2 Struktur

### 3.2.1 Übersicht

Abbildung 3.2 zeigt einen Überblick über die implementierten Klassen. Um das Diagramm überschaubar zu halten, wurden nur die Klassen eingezeichnet, die zusätzlich zum Erweiterungs-Template von RapidMiner :NC implementiert wurden. Außerdem wurden nur die notwendigen Enumerationen und Klassen für den NLP4J-Tagger aufgenommen, da andere Tagger analog strukturiert sind. Es folgt eine kurze Erklärung der verschiedenen Komponenten:

**Operatoren:** *Evaluator* und *Nlp4j\_tagger* sowie alle anderen Klassen, die von der RapidMiner-Klasse *Operator* erben, sind auch in RapidMiner als Operatoren vertreten. Alle Operatoren außer *Evaluator* beinhalten einen POS-Tagger und überbrücken

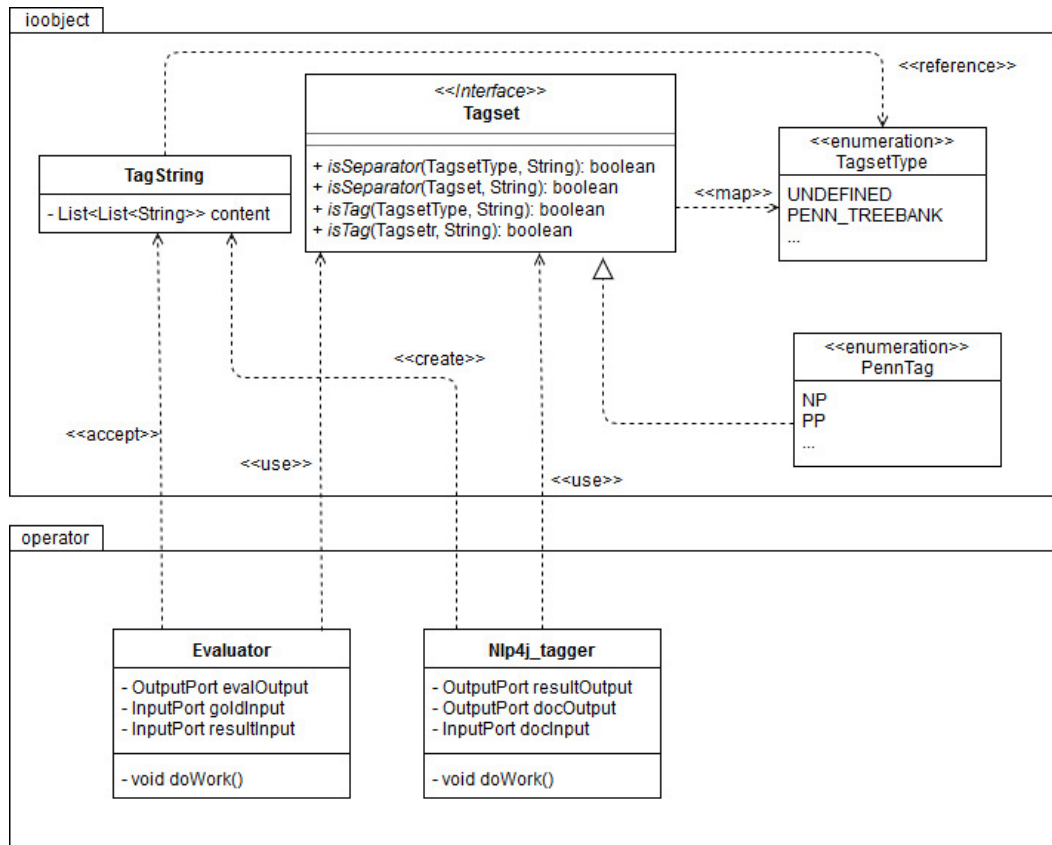


Abb. 3.2: gekürztes Klassendiagramm

sowohl Ein- als auch Ausgangsformate zwischen dem externen RapidMiner-Modell (*Document* oder *TagString*) und dem Tagging-Algorithmus selbst. Der Evaluator berechnet die Unterschiedlichkeit zwischen zwei Ergebnissen, wobei eines der beiden bestenfalls vollständig korrekt ist. :TODO

**Tagsets:** Das Interface *Tagset* muss von jeder Tag-Enumeration, wie zum Beispiel dem der Penn-Treebank implementiert werden. Zusätzlich bietet es statische Methoden, die die Werte der Enumeration *TagsetType* auf die korrespondierenden Enumerationen abbilden, sowie Abfragen über deren Werte anbieten (hierzu wird der Typ sowie ein Tag verlangt). Die implementierenden Tagsets führen alle Part-of-Speech-Tags auf und liefern zusätzlich die Information, ob das Tag in der Datenstruktur *TagString* (s.u.) Zeilen abbricht. Auf diese Weise ist es leicht, das Projekt um ein neues Set zu erweitern.

**TagString:** Diese Klasse bietet, wie in Abschnitt 3.1.2 angesprochen, ein einheitliches Format für POS-Tags. Sie beschreibt den Tagset-Typen sowie die Zahl an N-Besten Tags pro Token (1 für First-Best) und strukturiert die Tags in Zeilen, die immer dann enden, wenn das letzte Tag ein *Separator* ist. :TODO

## 3.3 Evaluation

Um Ergebnisse hinsichtlich ihrer Qualität zu prüfen, muss mit Goldstandards wie z.B. dem WSJ-Corpus :NC verglichen werden. Der Evaluationsoperator muss die Formate dieser Standards annehmen und interpretieren können. Neben dem eigentlichen Evaluieren ist also auch das *Parsen* eingehender Texte eine wichtige Aufgabe des Operators.

### 3.3.1 Parsing

Verschiedene Formate kodieren POS-Tags unterschiedlich. Eine einfache Notation ist die bereits bekannte Variante, bei der nach jedem Token ein „\“ und dann das Tag folgt. Andere Notationen kodieren zusätzlich Satzstrukturinformationen, sind also mittels Klammern geschachtelt, um z.B. Teilsätze zu markieren. Hier gibt es kein Symbol, das eindeutig ein POS-Tag ankündigt. Um die Parser-Methode des Evaluationsoperators präziser arbeiten können zu lassen, wurde ein Parameter hinzugefügt, in dem man das Format wählen kann. Es wurden zwei Modi für den Parser implementiert:

**Backslash-Notation:** In der oben genannten Notation, in der ausschließlich POS via „\“ markiert werden, ist Parsing einheitlich. Der Text wird an jedem Leerzeichen gespalten und jeder entstehende Substring am „\“. Das POS-Tag findet sich dann im zweiten String, der bei der zweiten Spaltung entsteht.

**"None":** Falls die Notation unbekannt ist, versucht der Parser, den Text an allen sinnvollen Symbolen, insbesondere dem Leerzeichen zu Trennen. Alle Substrings werden dann überprüft, ob sie ein POS-Tag sind.

Zusätzlich kann die Option „Ignore Brackets“ gewählt werden, falls Klammern für die Formatierung des einzulesenden Textes verwendet wurden. diese werden dann ignoriert.

Als Ergebnis gibt der Parser das Format TagString aus, das dann weiter verwendet werden kann. Der Tagset-Typ des TagStrings muss via Parameter angegeben werden.

## 3.4 Conclusion



# Concepts

Die folgenden POS-Tagger wurden implementiert:

Tagger (Training Korpus)	Performance (Korpus)
NLP4J	:TODO
LingPipe	:TODO
FastTag	-

**Tab. 4.1:** Liste der Implementierten Tagger. Performance laut eigenen Angaben.  
\* : Performance aus :NC

Die Performance in Tab. ?? bezieht sich auf eigene Angaben der Entwickler. Mit dem implementierten Evaluationsoperator sind wir allerdings in der Lage, die Tagger an einem eigenen annotierten Goldstandard-Korpus zu testen. Die folgenden Abschnitte diskutieren die Auswahl eines passenden Korpus und die Performance der Tagger auf diesem.

## 4.1 Goldstandard-Korpus

Ein annotierter Korpus muss dem Tagset der einzelnen Tagger entsprechen. Dies erweist sich als schwierig, da jeder Tagger u.U. Zeichen in unterschiedliche Token-Ketten spaltet und anders Kodiert. Es macht also Sinn, den Korpus und seine Tags auf den Tagger anzupassen, sofern das möglich ist, ohne den Informationsgehalt des Korpus zu ändern.

## 4.2 Concepts Section 2

## 4.3 Concepts Section 3

## 4.4 Conclusion



## Conclusion

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

### 5.1 System Section 1

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie

„Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 5.2 System Section 2

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander

stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst

viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 5.3 Future Work

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“?

Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.





# Abbildungsverzeichnis

3.1	Datenfluss in einem typischen Tagging- und Evaluationsprozess . . . .	5
3.2	gekürztes Klassendiagramm . . . . .	7



# Tabellenverzeichnis

4.1	Liste der Implementierten Tagger. Performance laut eigenen Angaben. *	
	: Performance von :NC . . . . .	9



# Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

*Bayreuth, 23.April 2019*

---

Philipp Scholz

