Bachelorarbeit

im Studiengang Computerlinguistik

an der Ludwig-Maximilians-Universität München

Fakultät für Sprach- und Literaturwissenschaften

Koreferenzen im Nachlass von Ludwig Wittgenstein für die FinderApp WITTFind

vorgelegt von
Oksana Budurova

Betreuer:	Dr. Maximilian Hadersbeck
Prüfer:	Dr. Maximilian Hadersbeck
Bearbeitungszeitraum:	23. März - 28. Mai 2018

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, den 28. Mai 2018

.....
Oksana Budurova

Abstract

Die vorliegende Arbeit beschäftigt sich mit der Implementierung einer Erweiterung der semantischen Suche der FinderApp WITTFind. Die App ist eine Suchmaschine, die für Ludwig Wittgensteins Werke entwickelt wurde. Sie soll eine Unterstützung derjenigen sein, die den Nachlass Ludwig Wittgensteins untersuchen und erforschen wollen. Das Tool enthält sowohl eine regelbasierte als auch eine semantische Suche. Das semantische Suchen wird nun, basierend auf Koreferenzen, um eine Kategorie erweitert. Mit dieser Kategorie können sowohl Personennamen als auch Pronomen, die sich auf Personen beziehen, gefunden werden. Dadurch wird der Zugriff auf weitere relevante Stellen des Korpus ermöglicht, die zuvor nicht durch die Suchmaschine erfasst werden konnten. Dadurch kann Textbedeutung genauer dargestellt werden.

This Bachelor's thesis deals with the implementation of a semantic search functionality for the application WITTFind. It was developed specifically for the corpus of Ludwig Wittgenstein's works with the purpose of analyzing his inheritance. The tool contains a rule based search option and a semantic one. This semantic search option now will be extended by a section for coreferences in Wittgenstein's corpus. With this new addition of coreferences, pronouns corresponding to persons can be found. It provides the opportunity to display text meaning in a more accurate way.

Представлена бакалаврська робота була написана при Центрі обробки інформації та мови (Centrum für Informations- und Sprachverarbeitung (CIS)) в Університеті Людвіга Максиміліана в місті Мюнхен, Німеччина. Вона займається розробкою семантичного пошуку для веб-програми WITTFind. Ця програма є інструментом для аналізу спадку австро-англійського філософа Людвіга Віттгенштайна (Ludwig Wittgenschtein). ВіТТФайнд (WiTTFind) є онлайн пошуковою системою з семантичною та схематичною опціями. Пошук власних імен в роботах зі спадку Вітгенштейна вже можливий. Метою даної роботи є створення анотацій для займенників з посиланням на власні імена, до яких вони відносяться, та таким чином розширення можливостей семантичного пошуку для кращої обробки корпусу та показу всіх релевантних місць текстів Людвіга Вітгенштейна. Розпізнані займенникові посилання в прикладах походять зі спадку Віттгенштайна.

Inhaltsverzeichnis

1 Einleitung

Ein Satz hat Sinn nur im Kontext. [1]

Ludwig Wittgenstein (1889 - 1951) war einer der bedeutendsten Philosophen des 20. Jahrhunderts. Sein Denken beeinflusste sowohl Psychologie, Architektur als auch Musik. Sein Nachlass enthält 20.000 Seiten teilweise unveröffentlichter Manuskripte und Typoskripte. Im Jahre 2000 veröffentlichten die Wittgensteinarchive an der Universität Bergen (WAB) den Nachlass in einer CD-ROM Edition als Bergen Electronic Edition (Wittgenstein et al. 1998). Im Jahr 2009 stellte WAB weitere 5.000 Seiten des Nachlasses auf der Webseite Wittgenstein Source zur Verfügung (Pichler et al. [2009]).

Im Rahmen des Digital-Humanities-Projekts „Wittgenstein in Co-Text“ wird seit acht Jahren in Zusammenarbeit mit dem WAB am Centrum für Informations- und Sprachverarbeitung (CIS) der LMU München die FinderApp „WiTTFind“ entwickelt. Die App ist eine, für Ludwig Wittgensteins Werke entwickelte, Suchmaschine. Sie ist von großer Bedeutung für diejenigen, die den Nachlass Ludwig Wittgensteins untersuchen und erforschen. Das Tool enthält sowohl eine regelbasierte, als auch eine semantische Suche. Eine eigens implementierte Web-Oberfläche bietet eine einfache Bedienbarkeit.

Im Mittelpunkt dieser Arbeit steht die Koreferenzauflösung. Sie ist von besonderer Relevanz, sowohl in der Computerlinguistik, als auch im Natural Language Processing (NLP). Dieses Gebiet ist wichtig für das computerlinguistische Forschen, insbesondere unter den Aspekten Information Retrieval, Information Extraction, Sentiment Analysis, maschinelle Übersetzung, Textzusammenfassung und Dialogmodellierung. Sie wird in der Preprocessingphase eingesetzt. Damit werden die Bezüge innerhalb des Textes suchbar gemacht. Dabei stellt die Koreferenzauflösung eine Herausforderung dar. Die neueste Methode handelt von semantischen sowie syntaktischen Filtern. Seit den Message Understanding Conferences (MUC) der 1990er Jahre bis einschließlich heute, haben alle wichtigen NLP Konferenzen dieses Thema behandelt und es wurden in entsprechende Publikationen veröffentlicht. Am häufigsten wird mit englischen Daten geforscht. Die Modelle darüber hinaus immer mehr an die Spezifikationen anderer Sprachen angepasst, insbesondere an die deutsche Sprache. Das Ziel dieser Arbeit ist es, durch die Koreferenzauflösung und durch die Verbesserung der XML-Edition des Nachlasses, die Erkennung von Personen in der WiTTFind zu optimieren. Für die Lösung dieser Aufgaben werden XML-Elemente bzw. XML-Attribute mit Hilfe der XML-Bibliothek `lxml` für Python geparst und modifiziert. Alle Beispiele, die im Laufe dieser Arbeit gezeigt werden, stammen aus Ludwig Wittgensteins Nachlass.

In dieser Bachelorarbeit sollen drei grundsätzliche Fragen bearbeitet werden. Zunächst soll geklärt werden, ob die Koreferenzen mit Hilfe des Koreferenzauflösungstools CorZu (Coreference Resolver for German from Zurich) erkannt werden können. Außerdem soll untersucht werden, wie diese Informationen für die Erweiterung der Edition des Nachlasses benutzt werden können. Abschließend wird betrachtet, ob die XML-Elemente aus Wittgensteins Bemerkungen angepasst, und modifiziert werden können.

Zu Beginn soll ein Überblick über die relevanten Arbeiten verschafft werden. Anschließend wird auf die Funktionalität der Suchmaschine WITTFind eingegangen. Ebenso wird auf die Dissertationsarbeit zurückgegriffen, welche das Koreferenzauflösungsmodell eingeführt, und dessen Anwendung auf die deutsche Sprache exploriert hat und somit die Basis für alle in dieser Arbeit gezeigten Entwicklungen darstellt. Diese Bausteine sind

[1] Wittgenstein, *Tractatus logico philosophicus*

von großer Bedeutung für eine effiziente Implementierung der Koreferenztags in die XML-Edition des Nachlasses. In Kapitel 3 wird die Struktur der Daten Wittgenstein Nachlasses beschrieben. Danach wird auf den Einsatz des Tools CorZu eingegangen, welches eine Konvertierung der Daten notwendig macht. Abschließend wird exemplarisch das Ergebnis des Koreferenzauflösungsmodells gezeigt. Das Kapitel 4 befasst sich mit der Zusammenführung zweier XML-Dateien. Die ursprüngliche Datei mit Wittgensteins Bemerkungen, soll durch die Koreferenztags modifiziert werden. Danach werden die Vorschläge des Koreferenzauflösungsmodells mit Beschränkung auf die Erkennung der Pronomen evaluiert, welche sich auf Personennamen beziehen.Im Ausblick werden weitere Möglichkeiten für die Verfeinerung der Edition des Wittgenstein Nachlasses präsentiert. In der Zusammenfassung wird noch mal abschließend diese Arbeit reflektiert.

2 Relevante Arbeiten

Der folgende Teil stellt Arbeiten und Artikel vor, die für dieses Thema relevant sind.

2.1 WITTFind Suchmaschine

Die Artikel von Hadersbeck et al. [2014] und [Hadersbeck et al., 2012] präsentieren die Suchmaschine WiTTFind. Sie bietet neue Möglichkeiten, Werke Wittgensteins zu untersuchen, was mittels Alternativen wie z.B. Google Books nicht immer möglich ist.

In Verbindung mit elektronischen Lexika und anderen computerlinguistischen Tools, ermöglichen lokale Grammatiken lemmatisierte, semantische und syntaktische Suchen. Die Ergebnisse werden vom Browser im HTML-Format dargestellt. Faksimile von entsprechenden Segmenten werden gezeigt. Seit dem Jahr 2010 entwickeln die Wittgensteinarchive an der Universität Bergen (WAB), in Kooperation mit dem CIS, in der Forschungsgruppe "Wittgenstein in Co-Text", die oben genannte Suchmaschine WiTTFind.

Das elektronische Vollformlexikon CISLEX wird im Bericht von Langer et al. [1996] präsentiert. Es ist eines der größten Deutschlands und wurde über 20 Jahre lang am CIS entwickelt. Seine Anotationen wurden den Wörter des Nachlasses hinzugefügt und in einem separaten Lexikon WiTTLex gespeichert. Jeder Eintrag ist im DELA-Format und besteht aus Vollform, Lemma, lexikographischer Form, Inflektion und semantischer Information. WiTTLex erlaubt außerdem eine lemmatisierte Suche: Wird das Wort `denken` gesucht, so werden zusätzlich zur Grundform auch die morphologischen Varianten `dachte` und `gedacht` angezeigt. Auch eine inverse lemmatisierte Suche ist möglich: Hier führt die Suchangabe `dachte` zum Ausgangslemma `denken`.

Jedes Wort beinhaltet einen Hinweis auf die lexikographische Form. Diese Form wird als Placeholder verwendet. Sucht man z.B. `die <ADJ> Farbe`, werden alle Sätze mit dem Artikel `die`, gefolgt von einem beliebigen Adjektiv und dem Nomen `Farbe` gefunden. Anschließend werden die Daten mit dem Wortartentreetagger getaggt, um Ambiguitäten zu vermeiden. In WiTTLex werden Nomen und Adjektive semantisch klassifiziert. Es gibt 11 Klassen für Nomen und Adjektive. Sie können in WiTTFind verwendet werden. Die Abbildung 2.1 und die Abbildung 2.3 zeigen die Häufigkeit des Vorkommens der semantischen Klassen im Nachlass. Als nächstes werden semantische Klassen für Farben implementiert. In der Tabelle 2.2 werden die Suchmöglichkeiten des Farbvokabulars Wittgensteins präsentiert.

Partikelverben verursachen ein Problem bei ihrer Erkennung, weil Partikel getrennt von den Stammverben im Satz vorkommen können. Beispiel: `Wir denken nie darüber nach...` Das dieses Problem behebende Tool wurde von Volos [2013] und Symonenko [2015] in WiTTFind implementiert. Es erlaubt, Verbpartikel als eben solche, statt als Präpositionen, zu erkennen. Die Bachelorarbeiten lösen das Problem der Mehrdeutigkeit von Verbpartikeln in Ludwig Wittgensteins Nachlass. Zu diesem Zweck hat Frau Volos [2013] in ihrer Bachelorarbeit mit Hilfe des Unitex Tools eine lokale Grammatik in Form eines Graphen erstellt, die Partikelverbkonstruktionen erkennen kann. In der Arbeit von Frau Symonenko [2015] wird diese Grammatik überarbeitet, weiterentwickelt und durch neue Erkennungsmuster ergänzt. Danach wird der dadurch entstandene Graph beim graphischen Finden als Suchanfrage für die Suchmaschine WiTTFind verwendet. Paumier et al. [2006] beschreiben die Vorteile des Einsatzes der Unitex Tools im Bereich Digital Humanities.

Name	Tag	Translation	Occurrences
Menschen	<HUM>	humans	140
Tiere	<T>	animals	96
Pflanzen	<PF>	plants	26
Objekte	<OBJ>	objects	1402
Ereignisse	<ER>	events	589
Zustände	<ZU>	states	51
Eigenschaften	<EIG> p	properties	236
Temporalia	<TEMP>	time	49
Eigennamen	<EN>	proper names	60
Numeralia	<NUM>	number	47
Diversa	<SONST>	other	713

Abbildung 2.1: Semantische Klassen von Nomen

Name	Tag	Translation	Occurrences
Grundfarbe	<Grundfarbe>	basic colour	454
Zwischenfarbe	<Zwischenfarbe>	intermediate colour	301
Transparenz	<Transparenz>	transparency	105
Glanz	<Glanz>	gloss	2
Farbigkeit	<Farbigkeit>	colourness	29

Abbildung 2.2: Semantische Klassen von Farben

Name	Tag	Translation	Occurrences
Farben	<COL>	colour	974
Numeralia	<NUM>	number	1258
Relation	<REL>	relation	2517
Eigennamen	<EN>	proper names	17
Temporalia	<TEMP>	time	619
Evaluation	<EVAL>	evaluation	1732
Zustände	<ZU>	states	6629
Komparativa	<KOMP>	compariative	2080
Stilistika	<STIL>	style	1917
Eigenschaft	<EIG>	property	382
Ereignisse	<ER>	proppperty	187

Abbildung 2.3: Semantische Klassen von Adjektiven

2.2 Koreferenzauflösung

Die hier beschriebene Arbeit ist eine Dissertation Tuggener [2016] von Don Tuggener (2016) an der Universität Zürich mit dem Titel „Incremental Coreference Resolution for German“. Die Arbeit handelt von Koreferenzauflösung und deren Anwendung auf die deutsche Sprache. Hier werden Lösungen sowie besondere Probleme bei der Verarbeitung von deutschen Pronomen dargestellt. Darüber hinaus wird eine Menge von neuen Merkmalen deutscher Pronomen eingeführt. Das Koreferenzmodell setzt Semantik ein, um die Kompatibilität eines Antezedenten zum darauf folgenden Pronomen zu bestimmen. Zu diesem Zweck werden Vektorrepräsentationen, mit syntaktischen Koreferenzprofilen von Wörtern verglichen.

Besonderheiten von deutschen Pronomen

Bestimmte deutsche Pronomen sind bezüglich ihrer morphologischen Eigenschaften unspezifisch wenn sie ohne Kontext betrachtet werden. Der Kontext ist wichtig, um potenzielle Antezedenten zu untersuchen. Im Vergleich zum Englischen ist im Deutschen die Menge der Pronomen größer. Dadurch wird die Wahrscheinlichkeit der Auswahl eines falschen Referenten erhöht. Zum Beispiel kann sich *sein* sowohl auf maskuline, als auch auf neutrale Nomen beziehen. Das kann zu einem falschen Mapping führen. Ein weiteres Beispiel ist, dass sich bestimmte deutsche Pronomen sowohl auf animierte, als auch auf unanimierte Entitäten beziehen: *er* referenziert auf eine männliche Person sowie ein unanimiertes Nomen wie *der Satz, der Sinn* usw.

Terminologie

Korefernzen von Aussagen und Entitäten können auf unterschiedliche Weise definiert werden. Die Terminologie im betrachteten Bereich ist nicht klar definiert. Da unterschiedliche Forscher unterschiedliche Bezeichnungen für diese sprachlichen Phänomene verwenden,sind Koreferenz, Wortsubstitution und Anaphorizität kombinierbar. Eine Anaphora stützt sich bei der Interpretation auf die vorherige Aussage. Koreferenz dagegen bedeutet, dass sich zwei Aussagen auf eine, zu Grunde liegende, Entität beziehen.

Koreferenz ist ein Phänomen der natürlichen Sprache. Unterschiedliche linguistische Formen werden im Diskurs mit dem Hinweis auf die gleiche extra-linguistische Entität benutzt. Zunächst wird beispielsweise *Lev Tolstoi* genannt. Anschließend wird eine nominale Beschreibung *einer der wichtigsten Schriftsteller Russlands* benutzt. Danach folgt das Pronomen *er*. Alle Aussagen beziehen sich auf die gleiche Person. Die linguistische Form ändert sich, wenn diese Aussagen nacheinander benutzt werden.

Pronominalisierung ist eine besondere Form der Substitution. Das verursacht ein Problem bei der Sprachverarbeitung, insbesondere innerhalb der FinderApp WiTTFind. Die Suchmaschine WiTTFind wird nur Entitäten finden, welche mit der Suchangabe übereinstimmen. Das Ziel der Koreferenzauflösung ist die Identifizierung und das Mapping der unterschiedlichen Formen zu einem Identifier, um das Vorkommen der Entitäten im Diskurs zu verfolgen. Die Wiederverwendung der Konzepte identifiziert die Semantik des Textes. In dieser Arbeit wird die Koreferenz als ein allgemeiner Begriff verwendet. Das erste Vorkommen eines Nomens, welches ein Pronomen identifiziert, ist ein Antezedent. Das folgende Pronomen ist eine Anaphora.

Verwendete Technologie

Das Tool basiert auf dem ParZu Parser (Zurich Dependency Parser for German). Sennrich et al. [2009] beschreiben die Entwicklung des Parsers für die deutsche Sprache. Sennrich et al. [2013] berichten über die Ergebnisse, nachdem sowohl Wortarttagger (POS-Tagger), als auch die morphologische Analyse verwendet wurden. Der Parser analysiert die linguistische Struktur der Sätze. Er kann auch sowohl ein Subjekt als auch ein Objekt eines

Verbes feststellen. Dieses Modell ist ein Hybrid aus regelbasierter Grammatik und statistischem Modul, dass die wahrscheinlichste Analyse eines Satzes widergibt. Der Parser spezialisiert sich auf die deutsche Sprache. Das Korreferenzauflösungstool braucht eine dependenzgeparste deutsche Datei im CoNLL Format. In der letzte Zeile werden die Koreferenzmarkierungen hinzugefügt.

```
ich ich PRO PPER 1|Sg|_|Nom 12 subj _ (0)
```

Der Nutzen von Koreferenzauflösungen

Information Retrieval stützt sich auf die Koreferenzauflösung. Die Relevanz der Dokumente für eine Suchangabe basiert auf Wortfrequenzen. In dem Fall wird das Algorithmus TF-IDF verwendet. Wenn man nach Termfrequenzen sucht, findet man keine Koreferenzen bzw. Pronomen oder Substantivbeschreibungen, weil sie durch andere Formulierungen vertreten werden. Daher steigert sich die Termfrequenz nicht. Die Koreferenzauflösung kann die Relevanz der gelieferten Dokumente wesentlich verbessern.

Beim Template Filling Task im Bereich Information Extraction können die unspezifischen Pronomen in Teplate Slots nicht auf die entsprechenden Antezedenten zurückgeführt werden. Bei Relation Extraction Tasks stellen die extrahierten Pronomen, welche als Argumente des Verbes agieren, keinen Mehrwert für die weiteren Aufgaben dar.

In unterschiedlichen Sprachen haben die Pronomen unterschiedliche grammatikalische Genera. Einen Antezedenten zu definieren wäre hilfreich, damit das maschinelle Übersetzungssystem das richtige Pronomen in die Zielsprache einfügen kann. Bei Sentiment Analysis ohne Koreferenzen werden die Kontexte der gesuchten Entitäten vermisst, wenn die Entität durch ein Pronomen oder einen anderen Gattungsnamen ausgetauscht wird. Der Recall-Wert könnte nach der Koreferenzauflösung steigen.

3 Implementierung des Koreferenzauflösungstools CorZu

Im folgenden Abschnitt „Implementierung des Koreferenzauflösungstools CorZu“ wird dargestellt, welche Schritte nötig waren, um die Arbeit mit der Koreferenzauflösung zu beginnen. Das Ziel des Verfahrens ist vor allem einen Überblick über das Koreferenzauflösungstools CorZu zu geben.

3.1 Struktur der Daten des Wittgenstein Nachlasses

Der Nachlass enthält 20.000 Seiten Manuskripte und Typoskripte. Sie wurden digitalisiert und im XML-Format zur weiteren Editierung bereitgestellt. Es wurden 3 unterschiedliche Datentypen für WITTFind erzeugt. OA.xml-Dateien beinhalten, exakt wie im Dokument von Wittgenstein, Vorschläge zur Korrektur. In DIPLO.xml-Dateien wird versucht, Details möglichst genau mit Hilfe von XML zu beschreiben. NORM.xml-Dateien werden in eine, aus der Sicht der Experten und Editoren, gut lesbare Form gebracht. In diesem Kapitel wird mit dem normalisierten Manuskript 101 gearbeitet, weil es das Tagebuch Wittgensteins darstellt und damit zur Verwendung bei der Koreferenzauflösung geeignet ist.

Grundsätzlich wurden alle Editionstexte in Bemerkungen geteilt, welche als separate Einheiten agieren. Außerdem sind sie als einmalig gekennzeichnet. Die Bemerkungen werden als `<ab n= ...ana = ...>` - Elemente getaggt. Das Attribut `n` zeigt die Koordinaten im Facsimile und `ana` bezeichnet die Absatznummer. Es wurden 10 Bemerkungen ausgewählt, bei denen Pronomen und Eigennamen vorkommen. Hier ein Beispiel von einer vollständigen Bemerkung mit ihrer Bezeichnung:

```
<xml>
        <ab n="Ms-101,2r[2]" ana="abnr:5">
            <s n="Ms-101,2r[2]_1" ana="facs:Ms-101,2r abnr:5 satznr:21">Schlecht geschlafen (Ungeziefer).</s>
            <s n="Ms-101,2r[2]_2" ana="facs:Ms-101,2r abnr:5 satznr:22">Nachdem ich das Zimmer <lb/>gekehrt hatte marschierten wir zu ein paar
 alten Mörsern <lb/> &amp;amp; wurden im Gebrauch instruiert.</s>
            <s n="Ms-101,2r[2]_3" ana="facs:Ms-101,2r abnr:5 satznr:23">Furchtbar heiß.</s>
            <lb/>
            <s n="Ms-101,2r[2]_4" ana="facs:Ms-101,2r abnr:5 satznr:24">Das Essen ist uneßbar.</s>
            <s n="Ms-101,2r[2]_5" ana="facs:Ms-101,2r abnr:5 satznr:25">Werde vielleicht in Zukunft <lb/>außerhalb der Kaserne schlafen.</s>
            <s n="Ms-101,2r[2]_6" ana="facs:Ms-101,2r abnr:5 satznr:26">An <persName key="Pinsent, David Hume">David</persName>
               <lb/> geschrieben.</s>
            <s n="Ms-101,2r[2]_7" ana="facs:Ms-101,2r abnr:5 satznr:27">Sehne mich schon nach einem Brief <lb/>von ihm um das Gefühl des Kontakts
 mit <lb/>meinem früheren Leben nicht zu verlieren.</s>
            <s n="Ms-101,2r[2]_8" ana="facs:Ms-101,2r abnr:5 satznr:28">Noch <lb/>nicht gearbeitet.</s>
         </ab>

</xml>
```

Abbildung 3.1: Bemerkung Ms-101,2r[2]

3.2 Konvertierung der Daten für die Koreferenzauflösung

In der Arbeit wird ein Wrapper von Herrn Lechner [2016] in Form eines Pythonskriptes verwendet. Es kann eine reine Text-Datei mit XML-Tags markieren, die ihrerseits Koreferenzen im Text identifiziert. Es wurde auf der Basis von CorZu (Coreference Resolver for German from Zurich) von Tuggener [2016] erstellt.

Die ursprünglichen Daten liegen im XML-Format vor. Der nächste Schritt ist die automatische Konvertierung in Text. Das dazu dienende Skript in Python wird als CD beigelegt.

Bemerkung	id	Antezedent	Anaphora
Ms-101,21v[2], absn:24	0	key = wir	wir
	0	key = wir	uns
	1	key = mich	mich
	1	key = mich	mir
	1	key = mich	ich
	1	key = mich	meiner
	1	key = mich	ich

Tabelle 3.1: Koreferenz in Ms-101,21v[2], absn:24

3.3 Analyse der Koreferenzauflösung

In diesem Abschnitt wird exemplarisch gezeigt worauf sich die korrekt aufgelösten Koreferenzen, nach dem Benutzen des Koreferenztools, beziehen. Jedes Beispiel gliedert sich in drei Teile. Zunächst wird eine Bemerkung im Text-Format betrachtet. Sie wurde von einem Python-Skript aus der XML-Datei erstellt. Das Skript wird als CD beigelegt. Die erkannten Wörter sind fett markiert. Als nächstes wird die Ausgabe des Tools CorZu im XML-Format mit entsprechenden Tags und Attributen näher beleuchtet. Im Folgenden Abschnitt ist eine Tabelle, die semantische Beziehungen innerhalb der Bemerkung übersichtlich erläutert, dargestellt. Die Tags `<coref>` wurden von dem Koreferenzauflösungsmodell verteilt. Das Attribut `id` bezeichnet die Reihenfolge des Vorkommens der Koreferenz im Text. Das Attribut `key` zeigt den Antezedenten, d. h. das Satzmitglied, auf welches sich die Anaphora bezieht. Die Anaphora steht als Taginhalt zwischen den Klammern `>...<`.

3.3.1 Positive Ergebnisse

Ms-101,21v[2] Absatz 24 *13.9.14. Heute in aller Früh verließen* ***wir*** *das Schiff mit allem was darauf war. Die Russen sind* ***uns*** *auf den Fersen. Habe furchtbare Szenen miterlebt. Seit 30 Stunden nicht geschlafen; fühle* ***mich*** *sehr schwach und sehe keine äußere Hoffnung. Wenn es mit* ***mir*** *jetzt zu Ende geht so möge* ***ich*** *einen guten Tod sterben, eingedenk* ***meiner*** *selbst. Möge* ***ich*** *mich nie selbst verlieren.*

```
<Coref> /
13.9.14. Heute in aller Frueh verliessen
<coref id = "0" key = "wir">wir</coref> /
das Schiff mit allem war darauf war. Die Russen sind
<coref id = "0" key = "wir">uns</coref> auf den Fersen. /
Habe furchtbare Szenen miterlebt. Seit 30 Stunden nicht geschlafen; /
fuehle <coref id = "1" key = "mich">mich</coref> /
sehr schwach und sehend keine aeussere Hoffnung. /
Wenn es mit <coref id = "1" key = "mich">mir</coref> /
jetzt zu Ende geht so moege <coref id = "1" key = "mich">ich</coref> /
einen guten Tod sterben, eingedenk /
<coref id = "1" key = "mich">meiner</coref> selbst. /
Moege <coref id = "1" key = "mich">ich</coref> mich nie selbst verlieren./
</Coref>
```

Aus der Tabelle 3.1 wird ersichtlich, dass sowohl verschiedene Personalpronomen (`wir, uns, mich, mir, ich`), als auch Possessivpronomen (`meiner`) erkannt wurden. Das Ergebnis ist durchaus plausibel.

Bemerkung	id	Antezedent	Anaphora
Ms-101,11r[3] absn:16	0	key = ich	ich
	0	key = ich	Mein
	0	key = ich	ich
	0	key = ich	mir

Tabelle 3.2: Koreferenz in Ms-101,11r[3] absn:16

Ms-101,11r[3] Absatz 16 *29.8.14. Jede Nacht stehe* **ich** *auf der Kommandobrücke bis etwa 312 a.m.* **Mein** *Vorhaben der vollkommenen Passivität habe* **ich** *noch nicht recht ausgeführt. Die Niedertracht der Kameraden ist* **mir** *noch immer schrecklich. Aber nur bei sich bleiben! Arbeite täglich etwas aber noch ohne rechten Erfolg. Obwohl schon manches aufdämmert.*

```
<Coref> 29.8.14. Jede Nacht stehe <coref id="0" key = ich>ich</coref> /
auf der Kommandobruecke bis etwa 31 2a.m.. /
<coref id="0" key = ich>Mein</coref> Vorhaben der vollkommenen /
Passivitaet habe <coref id="0" key = ich>ich</coref> /
noch nicht recht ausgefuehrt. Die Niedertracht der Kameraden /
ist <coref id="0" key = ich>mir</coref> noch immer schrecklich. /
Aber nur bei sich bleiben! Arbeite taeglich etwas aber /
noch ohne rechten Erfolg. Obwohl schon manches aufdaemmert./
</Coref>
```

Die Tabelle 3.2 verdeutlicht, dass dieses Modell eine hohe Perfomanz bei der Verknüpfung der Antezedenten und Anaforen zeigt, wenn die Pronomen in jedem nachfolgenden Satz vorkommen.

Ms-101,11r[2] Absatz 15 *26.8.14. Habe* **mir** *gestern vorgenommen keinen Widerstand zu leisten.* **Mein** *Äußeres sozusagen ganz leicht zu machen um* **mein** *Inneres ungestört zu lassen.*

```
<Coref> 26.8.14. Habe <coref id="0" key=mir >mir</coref>/
gestern vorgenommen keinen Widerstand zu leisten. /
<coref id="0" key=mir >Mein</coref> /
Aeusseres sozusagen ganz leicht zu machen um /
<coref id="0" key=mir >mein</coref> Inneres/
ungestoert zu lassen.</Coref>
```

Bemerkung	id	Antezedent	Anaphora
Ms-101,11r[2] absn:15	0	key = mir	mir
	0	key = mir	Mein
	0	key = mir	mein

Tabelle 3.3: Koreferenz in Ms-101,11r[2] absn:15

Die Verlinkungen in der Bemerkung Ms-101,11r[2] Absatz 15 (Tabelle 3.3) stimmen mit den vorher gezeigten Ergebnissen aus den Bemerkungen Ms-101,21v[2] in Absatz 24 und Ms-101,11r[3] Absatz 16 überein.

Ms-101,6r[2]et7r[1] Absatz 10 *18.8.14. Nachts um 1 werde* **ich** *plötzlich geweckt, der Oberleutnant fragt nach* **mir** *& sagt* **ich** *müsse sofort zum* **Scheinwerfer**. *„Nicht anziehen". Ich lief fast nackt auf die Kommandobrücke. Eisige Luft, Regen. Ich war sicher jetzt würde ich sterben. Setzte den Scheinwerfer in Gang & zurück mich anzukleiden. Es war falscher Alarm. Ich war furchtbar aufgeregt & stöhnte laut. Ich empfand die Schrecken des Krieges. Jetzt (abends) habe ich den Schreck schon wieder überwunden. Ich werde mein Leben mit aller Kraft zu erhalten trachten wenn ich nicht meinen gegenwärtigen Sinn ändere.*

```
<Coref> 18.8.14. Nachts um 1 werde <coref id="0" key = ich>ich</coref> /
ploetzlich geweckt, der Oberleutnant fragt nach /
<coref id="0" key = ich>mir</coref> &amp;amp; /
sagt <coref id="0" key = ich>ich</coref> muesse sofort zum /
<coref id="1" key="Scheinwerfer">Scheinwerfer</coref>. /
" Nicht anziehen ". <coref id="0" key = ich>Ich</coref> /
lief fast nackt auf die Kommandobruecke. Eisige Luft, Regen. /
<coref id="0" key = ich>Ich</coref> war sicher jetzt wuerde /
<coref id="0" key = ich>ich</coref> sterben. /
Setzte <coref id="1" key="Scheinwerfer">den Scheinwerfer</coref> /
in Gang \&amp;amp; zurueck <coref id="0" key = ich>mich</coref> anzukleiden. /
Es war falscher Alarm. <coref id="0" key = ich>Ich</coref> /
war furchtbar aufgeregt &amp;amp; stoehnte laut. /
<coref id="0" key = ich>Ich</coref> empfand die Schrecken des Krieges./
Jetzt ( abends) habe <coref id="0" key = ich>ich</coref> /
den Schreck schon wieder ueberwunden. /
<coref id="0" key = ich>Ich</coref> werde /
<coref id="0" key = ich>mein</coref> Leben mit aller Kraft /
zu erhalten trachten wenn <coref id="0" key = ich>ich</coref> /
nicht <coref id="0" key = ich>meinen</coref> gegenwaertigen Sinn aendere./
</Coref>
```

Bemerkung	id	Antezedent	Anaphora
Ms–101,6r[2]et7r[1]äbnr:10	0	key = ich	ich
	0	key = ich	mir
	0	key = ich	ich
	1	key = Scheinwerfer	Scheinwerfer
	0	key = ich	Ich
	0	key = ich	Ich
	0	key = ich	ich
	1	key = Scheinwerfer	den Scheinwerfer
	0	key = ich	mich
	0	key = ich	Ich
	0	key = ich	Ich
	0	key = ich	ich
	0	key = ich	Ich
	0	key = ich	mein
	0	key = ich	ich
	0	key = ich	meinen

Tabelle 3.4: Koreferenz in Ms-101,6r[2]et7r[1] abnr: 10

Die Tabelle 3.4 umfasst die Verlinkungskette in der Bemerkung Ms-101,6r[2]et7r[1] abnr:10. Es sollte auch nicht unerwähnt bleiben, dass das Modell die nominale Phrase

Bemerkung	id	Antezedent	Anaphora
Ms-101,8r[2] absn:12	0	key= mir	mir
	0	key = mir	Ich
	0	key = mir	meiner
	0	key = mir	Ich

Tabelle 3.5: Koreferenz in Ms-101,8r[2] absn:12

erkannt hat, insbesondere einen bestimmten Artikel mit dem Nomen `den Scheinwerfer`, obwohl diese 4 Sätze später erneut vorkommt. Die possessiven und personalen Pronomen werden korrekt verlinkt.

Ms-101,8r[2] Absatz 12 *Ob es jetzt für immer mit meinem Arbeiten aus ist?!! Das weiß der Teufel. Ob* `mir` *nie mehr etwas einfallen wird?* `Ich` *bin mit allen den Begriffen* `meiner` *Arbeit ganz & gar „unfamiliär". `Ich` sehe gar nichts!!!*

```
<Coref> Ob es jetzt fuer immer mit \underline{meinem} /
Arbeiten aus ist?!! Das weiss der Teufel. /
Ob <coref id="0" key=mir >mir</coref> nie mehr etwas einfallen wird? /
<coref id="0" key=mir >Ich</coref> bin mit allen den Begriffen /
<coref id="0" key=mir >meiner</coref> Arbeit ganz &amp; /
gar " unfamiliaer ". <coref id="0" key=mir >Ich</coref> sehe gar nichts!!!/
</Coref>
```

Das possessive Pronomen meinem wurde nicht verlinkt. Das Tool CorZu stellte fest, dass das Wort meinem inkompatibel sei. Die Verknüpfung wurde mit Beginn des drittes Satzes erstellt. Das personale Pronomen im Dativ `mir` agiert als Antezedent für die nachfolgenden Pronomen. Ungeachtet der Tatsache, dass das possessive Pronomen meinem nicht in die Verlinkungskette aufgenommen wurde, trägt das Ergebnis zur Koreferenzauflösung der Bemerkung bei.

Ms-101,2r[2] Absatz 5 *Schlecht geschlafen (Ungeziefer). Nachdem* ***ich*** *das Zimmer gekehrt hatte marschierten wir zu ein paar alten Mörsern & wurden im Gebrauch instruiert. Furchtbar heiß. Das Essen ist uneßbar. Werde vielleicht in Zukunft außerhalb der Kaserne schlafen. An* ***David*** *geschrieben. Sehne* ***mich*** *schon nach einem Brief von* ***ihm*** *um das Gefühl des Kontakts mit* ***meinem*** *früheren Leben nicht zu verlieren. Noch nicht gearbeitet.*

```
<Coref> Schlecht geschlafen ( Ungeziefer). /
Nachdem ich das Zimmer gekehrt hatte /
marschierten wir zu ein paar alten Moersern &amp;/
wurden im Gebrauch instruiert. Furchtbar heiss. /
Das Essen ist unessbar. Werde vielleicht in Zukunft /
ausserhalb der Kaserne schlafen. /
An <coref id="0" key="David">David</coref> geschrieben. /
Sehne <coref id="1" key="mich">mich</coref> /
schon nach einem Brief von <coref id="0" key="David">ihm</coref> /
um das Gefuehl des Kontakts mit <coref id="1" key="mich">meinem</coref> /
frueheren Leben nicht zu verlieren. /
Noch nicht gearbeitet.</Coref>
```

In der Tabelle 3.6 werden zwei Koreferenzen erkannt. `David` bezieht sich auf `ihm` und `mich` zeigt auf `meinem`. `ich` im zweiten Satz wurde nicht getaggt. Das Pronomen ich im zweiten Satz war nicht mehr als Antezedent verfügbar, wenn nach fünf Sätzen das Pronomen `mich` vorkommt. Die Abstandsbeschränkung des Tools beträgt drei Sätze (Tuggener [2016]).

Somit ist schlusszufolgern, dass das angebotene Modell nur bedingt verwendet werden kann. An dieser Stelle muss man besonders betonen, dass Abstandsbeschränkungen notwendig sind, um die Anzahl der Antezedentkandidaten zu reduzieren. Dieses Problem zu lösen bedarf weiterer Untersuchungen.

Bemerkung	id	Antezedent	Anaphora
Ms-101,2r[2] absn:5	0	key= David	David
	1	key = mich	mich
	0	key = David	ihm
	1	key = mich	meinem

Tabelle 3.6: Koreferenz in Ms-101,2r[2] absn:5

Ms-101,9r[2]et10r[1]et11r[1] Absatz 14 *Gestern ein furchtbarer Tag. Abends wollte* ***der Scheinwerfer*** *nicht funktionieren. Als* ***ich ihn*** *untersuchen wollte wurde* ***ich*** *von der Mannschaft durch Zurufe Grölen etc. gestört. Wollte* ***ihn*** *genauer untersuchen da nahm* ***ihn*** *der Zugsführer* ***mir*** *aus der Hand.* ***Ich*** *kann gar nicht weiter schreiben. Es war entsetzlich. Das Eine habe* ***ich*** *gesehen: Es ist nicht ein einziger anständiger Kerl in der ganzen Mannschaft. Wie aber soll* ***ich*** *mich in Zukunft zu dem Allen stellen. Soll* ***ich*** *einfach dulden? Und wenn* ***ich*** *das nicht tun will? Dann muß* ***ich*** *in einem fortwährenden Kampf leben. Was ist besser? Im 2. Fall würde* ***ich*** *mich sicher aufreiben. Im ersten vielleicht nicht. Es wird jetzt für mich eine enorm schwere Zeit kommen denn* ***ich*** *bin jetzt tatsächlich wieder so verkauft & verraten wie seinerzeit in der Schule in Linz. Nur eines ist nötig: Alles was einem geschieht betrachten können; sich sammeln! Gott helfe* ***mir!***

```
<Coref> Gestern ein furchtbarer Tag. /
Abends wollte <coref id="0" key="der Scheinwerfer">der Scheinwerfer</coref>/
nicht funktionieren. Als <coref id="1" key="ich">ich</coref>/
<coref id="0" key="der Scheinwerfer">ihn</coref> /
untersuchen wollte wurde <coref id="1" key="ich">ich</coref> /
von der Mannschaft durch Zurufe Groelenetc.. gestoert. /
Wollte <coref id="0" key="der Scheinwerfer">ihn</coref> /
genauer untersuchen da nahm /
<coref id="0" key="der Scheinwerfer">ihn</coref> /
der Zugsfuehrer <coref id="1" key="ich">mir</coref> /
aus der Hand. <coref id="1" key="ich">Ich</coref> /
kann gar nicht weiter schreiben. Es war entsetzlich. /
Das Eine habe <coref id="1" key="ich">ich</coref> gesehen: /
Es ist nicht ein einziger anstaendiger Kerl in der ganzen Mannschaft./
Wie aber soll <coref id="1" key="ich">ich</coref> /
mich in Zukunft zu dem Allen stellen. /
Soll <coref id="1" key="ich">ich</coref> nur einfach dulden? /
Und wenn <coref id="1" key="ich">ich</coref> das nicht tun will? /
Dann muss <coref id="1" key="ich">ich</coref> /
in einem fortwaehrenden Kampf leben. Was ist besser? /
Im 2. Fall wuerde <coref id="1" key="ich">ich</coref> /
mich sicher aufreiben. Im ersten vielleicht nicht. /
Es wird jetzt fuer <coref id="1" key="ich">mich</coref> /
eine enorm schwere Zeit kommen denn /
<coref id="1" key="ich">ich</coref> bin jetzt tatsaechlich/
wieder so verkauft &amp;amp; verraten wie seinerzeit /
in der Schule in Linz. Nur eines ist noetig: /
Alles was einem geschieht betrachten koennen; /
```

Bemerkung	id	Antezedent	Anaphora
	0	key = der Scheinwerfer	der Scheinwerfer
	1	key = ich	ich
	0	key = der Scheinwerfer	ihn
	1	key = ich	ich
	0	key = der Scheinwerfer	ihn
	0	key = der Scheinwerfer	ihn
	1	key = ich	mir
	1	key = ich	Ich
Ms-101,9r[2]et10r[1]et11r[1] abnr:14	1	key = ich	ich
	1	key = ich	ich
	1	key = ich	ich
	1	key = ich	ich
	1	key = ich	ich
	1	key = ich	ich
	1	key = ich	ich
	1	key = ich	mich
	1	key = ich	ich
	1	key = ich	mir

Tabelle 3.7: Koreferenz in Ms-101,9r[2]et10r[1]et11r[1] abnr:14

```
sich sammeln! Gott helfe <coref id="1" key="ich">mir</coref>!/
</Coref>
```

Die Tabelle 3.7 zeigt, dass die Antezedenten `ich` und `Scheinwerfer` korrekt festgestellt wurden. Dies verdeutlicht, dass das Modell die Pronomen der zuletzt erwähnten Entitäten richtig verlinkt, wenn die nominalen Phrasen und Pronomen in den Sätzen nacheinander folgen. Am Rande sei auch erwähnt, dass dieses Modell absichtlich reflexive Pronomen ignoriert. Aus diesem Grund wurde das Pronomen `mich` nicht als eine Anaphora für den Antezedent `ich` annotiert.

3.3.2 Aufgetretene Fehler

In diesem Abschnitt werden exemplarisch die fehlerhaften Ergebnisse der Koreferenzauflösung der CorZu (Coreference Resolution from Zurich) von Tuggener [2016] analysiert. Je mehr Sätze die Bemerkung enthält, desto höher ist die Wahrscheinlichkeit, dass die Tags falsch verteilt werden. Dabei spielen die morphologischen Eigenschaften der Antezedentkandidaten und Pronomen, sowie die Abstandsbeschränkungen, eine entscheidende Rolle. Die folgenden Beispiele illustrieren die auftretenden Probleme.

Ms-101,2r[3]et3r[1] Absatz 6 *Vorgestern beim* `Hauptmann` *gewesen. War sehr verdattert & stand nicht militärmäßig vor* `ihm`. `Er` *war etwas ironisch und* `mir` *nicht recht sympathisch. Resultat = 0. Heute kam es heraus daß* `ich` `Matura` *etc. gemacht hatte worauf eine ganze Reihe der Ein-jährigen* `mich` *mit Herr Kollege be-titelten & auf* `mich` *eindrangen* `ich` *solle doch* `mein` *Freiwilligenrecht geltend machen. Dies machte* `mir` *Spaß. (It bucked me up.) Gestern & heute starken Katarrh & oft Unwohlbe-finden. Manchmal ein wenig deprimiert. Traf heute in der Kantine* `einen` `Leutnant` `dem` *es auffiel daß* `ich` *dort zu Mittag aß* `er` *fragte* `mich` *sehr nett was* `ich` *im* `Zivil` *sei wunderte sich sehr daß* `sie` `mich` *nicht zu den einjährig Freiwilligen genommen hatten & war überhaupt sehr freundlich was* `mir` *sehr wohl tat.*

```
<Coref> /
Vorgestern beim <coref id="0" key= Hauptmann >Hauptmann</coref> /
gewesen. War sehr verdattert &amp;amp; /
```

```
stand nicht militaermaessig vor /
<coref id="0" key= Hauptmann >ihm</coref>. /
<coref id="0" key= Hauptmann >Er</coref> war etwas ironisch und /
<coref id="1" key=mir >mir</coref> nicht recht sympathisch. /
Resultat = 0. Heute kam es heraus dass /
<coref id="0" key= Hauptmann >ich Matura</coref>etc.. /
gemacht hatte worauf eine ganze Reihe der Einjaehrigen /
<coref id="1" key=mir >mich</coref> /
mit Herr Kollege betitelten \&amp; /
auf <coref id="1" key=mir >mich</coref> eindrangen/
<coref id="1" key=mir >ich</coref> solle doch /
<coref id="1" key=mir >mein</coref> Freiwilligenrecht geltend machen. /
Dies machte <coref id="1" key=mir >mir</coref> Spass. /
( It bucked me up.) /
Gestern \&amp; heute starken Katarrh &amp; /
oft Unwohlbefinden. Manchmal ein wenig deprimiert. /
Traf heute in der Kantine /
<coref id="2" key= einen Leutnant >einen Leutnant</coref> /
<coref id="2" key= einen Leutnant >dem</coref> /
es auffiel dass <coref id="3" key = ich>ich</coref> /
dort zu Mittag ass <coref id="2" key= einen Leutnant >er</coref> /
fragte <coref id="3" key = ich>mich</coref>
sehr nett was <coref id="3" key = ich>ich</coref> /
im <coref id="4" key="Zivil">Zivil</coref> sei /
wunderte sich sehr dass <coref id="4" key="Zivil">sie</coref> /
<coref id="3" key = ich>mich</coref> nicht /
zu den einjaehrig Freiwilligen genommen hatten \&amp; /
war ueberhaupt sehr freundlich was /
<coref id="3" key = ich>mir</coref> sehr wohl tat./
</Coref>
```

In der Tabelle 3.8 wird gezeigt, dass sowohl die Verknüpfungen `Hauptmann - ich Matura` als auch `Zivil - sie (Plural)` mangelhaft sind. Dies lässt sich aus dem Kontext erkennen. Das Modell hat in diesem Fall versagt. Hier sind die Gründe zu hinterfragen. Warum wurde im Satz `Heute kam es heraus daß <coref id="0"key= Hauptmann >ich Matura</coref>etc..` eine Phrase `ich Matura` einem Antezedenten `Hauptmann` zugeordnet? Zuerst wurden die Grenzen der nominalen Phrase `ich Matura` falsch gesetzt. Hier handelt es sich nur um ein Pronomen `ich`. Wären die Grenzen richtig festgelegt worden, hätte man vermeiden können, dass das Pronomen `ich` mit dem Nomen `Hauptmann` verlinkt wird, da diese morphologisch inkompatibel sind. Als nächstes wird der folgende Abschnitt betrachtet.

```
... was <coref id="3" key = ich>ich</coref> im /
<coref id="4" key="Zivil">Zivil</coref> sei /
wunderte sich sehr dass <coref id="4" key="Zivil">sie</coref> /
<coref id="3" key = ich>mich</coref> /
nicht zu den einjaehrig Freiwilligen genommen hatten ...
```

Weshalb hat die Verlinkung `Zivil - sie (Plural)` stattgefunden? Dem ersten Vorkommen von `Zivil`, folgt ein gemeinsames Auftreten von Präposition und bestimmtem Artikel im Dativ Singular `im`. Das Pronomen `sie` im folgenden Nebensatz tritt mit der Eigenschaft Plural auf, da das Verb `nehmen` im Plusquamperfekt ebenfalls im Plural vorkommt `(... sie ... genommen hatten ...)`. Morphologisch betrachtet gibt es klare Gegenargumente für eine Verlinkung. An dieser Stelle wäre es wichtig zu wissen, ob die Fehler beim Wortarttagging bzw. Parsing aufgetreten sind. Als werden die Wortvektoren betrachtet:

Bemerkung	id	Antezedent	Anaphora
	0	key= Hauptmann	Hauptmann
	0	key= Hauptmann	ihm
	0	key= Hauptmann	Er
	1	key = mir	mir
	0	key= Hauptmann	ich Matura
	1	key = mir	mich
	1	key = mir	mich
	1	key = mir	ich
	1	key = mir	mein
Ms-101,2r[3]et3r[1]	1	key = mir	mir
abnr:6	2	key= einen Leutnant	einen Leutnant
	2	key= einen Leutnant	dem
	3	key = ich	ich
	2	key= einen Leutnant	er
	3	key = ich	mich
	3	key = ich	ich
	4	key= Zivil	Zivil
	4	key= Zivil	sie
	3	key = ich	mich
	3	key = ich	mir

Tabelle 3.8: Koreferenz in Ms-101,2r[3]et3r[1] abnr:6

```
['24', 'im', 'in', 'PREP', 'APPRART', 'Dat', '26', 'pp','_', '-']

['25', 'Zivil', 'Zivil', 'N', 'NN', '_|Dat|_', '24', 'pn', '_', '(4)']

['31', 'sie', 'sie', 'PRO', 'PPER', '3|Pl|_|Nom', '39', 'subj','_', '(4)']
```

Das Beispiel zeigt, dass der Token 24 `im` keine Markierung `Sg` hat. Das führt dazu, dass der Token 25 `Zivil` keine Markierung `Sg` hat. Dies hat eindeutig eine Unspezifizierung zur Folge, welche später vom Modell falsch interpretiert wurde. Die Konsequenz ist, dass die zuletzt erwähnte Entität `Zivil` als Antezedent für `sie` verwendet wurde. Die morphologischen Eigenschaften wurden für kompatibel gehalten.

Ms-101,5r[2] Absatz 8 *16.8.14. Auf der „Goplana". Nochmals: Die Dummheit, Frechheit & Bosheit dieser Menschen kennt keine Grenzen. Jede Arbeit wird zur Qual. Aber* ***ich*** *habe heute schon wieder gearbeitet & werde* ***mich*** *nicht unterkriegen lassen.* ***Schrieb*** *heute eine Karte an den lieben David. Der Himmel beschütze* ***ihn*** *& erhalte* ***mir seine*** *Freundschaft! – Die Fahrt selbst entlang der Weichsel ist herrlich &* ***ich*** *bin in guter Stimmung.*

```
<Coref> 16.8.14. Auf der " Goplana ". Nochmals: /
Die Dummheit, Frechheit &amp; /
Bosheit dieser Menschen kennt keine Grenzen. /
Jede Arbeit wird zur Qual. /
Aber <coref id="0" key="ich">ich</coref> /
habe heute schon wieder gearbeitet &amp; /
werde <coref id="0" key="ich">mich</coref> /
nicht unterkriegen lassen. /
<coref id="1" key="Schrieb">Schrieb</coref> /
heute eine Karte an den lieben David. /
Der Himmel beschuetze <coref id="1" key="Schrieb">ihn</coref> &amp;/
erhalte <coref id="0" key="ich">mir</coref>/
```

```
<coref id="1" key="Schrieb">seine</coref> Freundschaft!/
- Die Fahrt selbst entlang der Weichsel ist herrlich &amp; /
<coref id="0" key="ich">ich</coref> bin in guter Stimmung./
</Coref>
```

Bemerkung	id	Antezedent	Anaphora
Ms-101,5r[2] abnr:8	0	key = ich	ich
	0	key = ich	mich
	1	key = `Schrieb`	`Schrieb`
	1	key = `Schrieb`	ihn
	0	key = ich	mir
	1	key = `Schrieb`	seine
	0	key = ich	ich

Tabelle 3.9: Koreferenz in Ms-101,5r[2] abnr:8

In der Bemerkung 3.9 sind folgende Fehler aufgetreten: Es wurde das Verb im Imperfekt `Schrieb` als Antezedent des Personalpronomens `ihm` erkannt. Der korrekte Antezedent wäre der Eigenname `David` gewesen. Die Wortvektoren zeigen die Eigenschaften der Token. z.B. das Token 1 `Schrieb` fälschlicherweise als `Nomen` statt als `Verb` getaggt ist, welches zu einer Kaskadierung der Fehler führt.

```
['1', 'Schrieb', 'Schrieb', 'N', 'NN', 'Masc|_|Sg', '0', 'root', '_', '(1)']

['5', 'an', 'an', 'PREP', 'APPR', 'Acc', '4', 'pp', '_', '-']

['6', 'den', 'die', 'ART', 'ART', 'Def|Masc|Acc|Sg', '8', 'det', '_', '-']

['7', 'lieben', 'lieb', 'ADJA', 'ADJA', 'Pos|Masc|Acc|Sg|_|', '8', 'attr',/
'_', '-']

['8', 'David', 'David', 'N', 'NE', 'Masc|Acc|Sg', '5', 'pn', '_', '-']

['4', 'ihn', 'er', 'PRO', 'PPER', '3|Sg|Masc|Acc', '3', 'obja', '_', '(1)']

['8', 'seine', 'seine', 'ART', 'PPOSAT', '_|_|Sg', '9', /
'det', '_', '(1)']
```

Ms-101,7r[2] Absatz 11 *`Der Leutnant` & `ich` haben schon oft über alles Mögliche gesprochen; ein sehr netter Mensch. `Er` kann mit den größten Halunken umgehen & freundlich sein ohne sich etwas zu vergeben. Wenn `wir` einen Chinesen hören so sind `wir` geneigt `sein` Sprechen für ein unartikuliertes Gurgeln zu halten. Einer der Chinesisch versteht wird darin die Sprache erkennen. So kann ich oft nicht den Menschen im Menschen erkennen etc. Ein wenig aber erfolglos gearbeitet.*

```
<Coref>/
<coref id="0" key="Der Leutnant">Der Leutnant</coref> &amp; /
<coref id="0" key="Der Leutnant">ich</coref> /
haben schon oft ueber alles Moegliche gesprochen; /
ein sehr netter Mensch. <coref id="0" key="Der Leutnant">Er</coref> /
kann mit den groessten Halunken umgehen &amp; /
freundlich sein ohne sich etwas zu vergeben. /
```

```
Wenn <coref id="1" key="wir">wir</coref> /
einen Chinesen hoeren so sind <coref id="1" key="wir">wir</coref>/
geneigt <coref id="0" key="Der Leutnant">sein</coref> Sprechen /
fuer ein unartikuliertes Gurgeln zu halten. /
Einer der Chinesisch versteht wird darin die Sprache erkennen. /
So kann ich oft nicht den Menschen im Menschen erkennen etc. /
Ein wenig aber erfolglos gearbeitet./
</Coref>
```

Bemerkung	id	Antezedent	Anaphora
Ms-101,7r[2] abnr:11	0	key = Der Leutnant	Der Leutnant
	0	key = `Der Leutnant`	ich
	0	key = Der Leutnant	er
	1	key = wir	wir
	1	key = wir	wir
	0	key = `Der Leutnant`	sein

Tabelle 3.10: Koreferenz in Ms-101,7r[2] abnr:11

In der Bemerkung Ms-101,7r[2] abnr:11 wurde fälschlicherweise der Antezedent `Leutnant` auf `ich` bezogen. Die nominale Phrase `Der Leutnant & ich` wurde nicht als eine Einheit in Plural erkannt. `&` wurde nicht als Konjunktion dargestellt. `ich` agiert wie eine Apposition bzw. Spezifikation. Da `einen Chinesen` nicht als nominale Phrase durch POS-tagging annotiert wurde, kann die Phrase nicht als ein Antezedentkandidat für das possessive Pronomen `sein`, berücksichtigt werden. Im Anschluss wurde der Antezedent `Leutnant` dem Pronomen `sein` zugeordnet. Der richtige Antezedent wäre das Nomen `Chinese` gewesen.

3.4 Manuelle Anpassung der Vektoren

Die Verbesserung des deutschen Dependenzparsers ist nicht Bestandteil der Ziel dieser Arbeit. Diese Frage bedarf eines separaten Projekts. Deswegen wurde eine manuelle Korrektur der Wortvektoren im Einzelfall vorgenommen. Die korrigierten Wortvektoren

- 2r_3manual_korrection.coref
- 5r_2manual_korrection.coref
- 7r_2manual_korrection.coref

sind als CD im Ordner `wordvector_correction` beigelegt. Sennrich and Haddow [2015] präsentieren die Dependenzlabels, welche zur Korrektur gedient haben.

Die Korrektur wurde erfolgreich umgesetzt. Das Tool CorZu hat in jedem Fall mit Hilfe der korigierten Wortvektoren die erwünschten Ergebnisse geliefert. Als Nächstes wird die Ausgabe des Tools im XML-Format dargestellt. Die koreferenzaufgelösten Dateien

- 2r_3VektCorrect.xml
- 5r_2VektCorrect.xml
- 7r_2VektCorrect.xml

können als CD im Ordner `xml_corefAfterVeCtCorrection` gefunden werden.

3.4.1 Ms-101,2r[3]et3r[1] Absatz 6

```
<Coref> Vorgestern beim /
<coref id="0" key="Hauptmann">Hauptmann</coref> /
gewesen. War sehr verdattert &amp; /
stand nicht militärmäßig vor /
<coref id="0" key="Hauptmann">ihm</coref>. /
<coref id="0" key="Hauptmann">Er</coref> /
war etwas ironisch und /
<coref id="1" key="mir">mir</coref> /
nicht recht sympathisch. Resultat = 0. /
Heute kam es heraus daß /
<coref id="1" key="mir">ich</coref> Matura etc.. /
gemacht hatte worauf eine ganze Reihe /
der Einjährigen <coref id="1" key="mir">mich</coref>/
mit Herr Kollege betitelten &amp; /
auf <coref id="1" key="mir">mich</coref> /
eindrangen <coref id="1" key="mir">ich</coref>/
solle doch <coref id="1" key="mir">mein</coref> /
Freiwilligenrecht geltend machen./
Dies machte /
<coref id="1" key="mir">mir</coref> Spaß./
( It bucked me up.) Gestern &amp; /
heute starken Katarrh &amp; /
oft Unwohlbefinden. Manchmal ein wenig deprimiert./
Traf heute in der Kantine /
<coref id="2" key="einen Leutnant">einen Leutnant</coref>/
<coref id="2" key="einen Leutnant">dem</coref> /
es auffiel daß <coref id="3" key="ich">ich</coref>/
dort zu Mittag aß /
<coref id="2" key="einen Leutnant">er</coref>/
fragte <coref id="3" key="ich">mich</coref>/
sehr nett was <coref id="3" key="ich">ich</coref>/
im Zivil sei wunderte sich sehr daß sie /
<coref id="3" key="ich">mich</coref> /
nicht zu den einjährig Freiwilligen/
genommen hatten &amp; war überhaupt /
sehr freundlich was /
<coref id="3" key="ich">mir</coref> /
sehr wohl tat.</Coref>
```

3.4.2 Ms-101,5r[2] Absatz 8

```
<Coref> 16.8.14. Auf der "Goplana"./
Nochmals: Die Dummheit, Frechheit &amp;/
Bosheit dieser Menschen kennt keine Grenzen. /
Jede Arbeit wird zur Qual. /
Aber <coref id="0" key="ich">ich</coref> /
habe heute schon wieder gearbeitet &amp;/
werde <coref id="0" key="ich">mich</coref> /
nicht unterkriegen lassen. /
Schrieb heute eine Karte an den lieben/
<coref id="1" key="David">David</coref>./
```

```
Der Himmel beschütze /
<coref id="1" key="David">ihn</coref> &amp; /
erhalte <coref id="0" key="ich">mir</coref> /
<coref id="1" key="David">seine</coref>/
Freundschaft! /
- Die Fahrt selbst entlang der Weichsel /
ist herrlich &amp;/
<coref id="0" key="ich">ich</coref> /
bin in guter Stimmung.</Coref>
```

3.4.3 Ms-101,7r[2] Absatz 11

```
<Coref> /
<coref id="0" key="Der Leutnant">Der Leutnant</coref> /
&amp; <coref id="1" key="ich">ich</coref> /
haben schon oft über alles Mögliche gesprochen; /
ein sehr netter Mensch. /
<coref id="0" key="Der Leutnant">Er</coref> /
kann mit den größten Halunken umgehen &amp; /
freundlich sein ohne sich etwas zu vergeben. /
Wenn <coref id="2" key="wir">wir</coref> /
<coref id="3" key="einen Chinesen">einen Chinesen</coref>/
hören so sind/
<coref id="2" key="wir">wir</coref>/
geneigt /
<coref id="3" key="einen Chinesen">sein</coref> /
Sprechen für ein unartikuliertes Gurgeln /
zu halten. Einer der Chinesisch versteht /
wird darin die Sprache erkennen. /
So kann <coref id="1" key="ich">ich</coref> /
oft nicht den Menschen im Menschen erkennen etc./
Ein wenig aber erfolglos gearbeitet.</Coref>
```

3.5 Frequenz der gefundenen Koreferenzen

Die hier präsentierten Grafiken wurden mit Hilfe der Bibliothek `plotly` für die Programmiersprache Python erstellt. Das Python-Skript, das die Koreferenzfrequenzen in Bar Charts umwandelt, wird als CD im Ordner `stat` hingefügt. Aus dieser Untersuchung ergibt sich, dass vor der Korrektur der Wortvektoren bei 10 Bemerkungen 88 Koreferenzen festgestellt wurden, welche in der Abbildung 3.2 dargestellt werden. Aus der Grafik lässt sich herauslesen, dass das im Manuskript am häufigsten vorkommende Pronomen `ich` ist.

Die folgende Liste enthält die fehlerhaften Koreferenzen, welche nach der Wortvektorenoptimierung behoben wurden.

```
ich , ich , 26
mir , ich , 6
mich , ich , 5
mir , mir , 4
wir , wir , 3
ich , mir , 3
mein , mir , 3
ihn , der Scheinwerfer , 3
mich , mir , 2
```

```
ich , mich , 2
mein , ich , 2
mich , mich , 2
mir , mich , 1
ihm , Hauptmann , 1
Scheinwerfer , Scheinwerfer , 1
ich Matura , Hauptmann , 1
sein , Der Leutnant , 1
ihn , Schrieb , 1
uns , wir , 1
meiner , mir , 1
den Scheinwerfer , Scheinwerfer , 1
dem , einen Leutnant , 1
Schrieb , Schrieb , 1
Hauptmann , Hauptmann , 1
David , David , 1
einen Leutnant , einen Leutnant , 1
er , einen Leutnant , 1
der Scheinwerfer , der Scheinwerfer , 1
meinen , ich , 1
meiner , mich , 1
Der Leutnant , Der Leutnant , 1
sie , Zivil , 1
ihm , David , 1
er , Hauptmann , 1
ich , Der Leutnant , 1
meinem , mich , 1
er , Der Leutnant , 1
Zivil , Zivil , 1
seine , Schrieb , 1
```

Nach der Wortvektorenkorrektur wurden 88 Koreferenzen erneut annotiert. Abbildung 3.3 zeigt die optimierten Ergebnisse bei der Koreferenzauflösung. Die folgende Liste zeigt die festgestellten Koreferenzfrequenzen nach der Wortvektorenkorrektur. Die Koreferenz `ich - ich` kommt am häufigsten vor.

```
ich , ich , 28
mir , ich , 6
mich , ich , 5
ich , mir , 4
mir , mir , 4
ihn , der Scheinwerfer , 3
mein , mir , 3
wir , wir , 3
David , David , 2
ich , mich , 2
mich , mich , 2
mein , ich , 2
mich , mir , 2
der Scheinwerfer , der Scheinwerfer , 1
seine , David , 1
einen Chinesen , einen Chinesen , 1
er , einen Leutnant , 1
```

```
ihm , David , 1
Hauptmann , Hauptmann , 1
meiner , mir , 1
einen Leutnant , einen Leutnant , 1
ihm , Hauptmann , 1
ihn , David , 1
meinen , ich , 1
er , Hauptmann , 1
er , Der Leutnant , 1
sein , einen Chinesen , 1
meiner , mich , 1
Der Leutnant , Der Leutnant , 1
uns , wir , 1
dem , einen Leutnant , 1
mir , mich , 1
den Scheinwerfer , Scheinwerfer , 1
meinem , mich , 1
Scheinwerfer , Scheinwerfer , 1
```

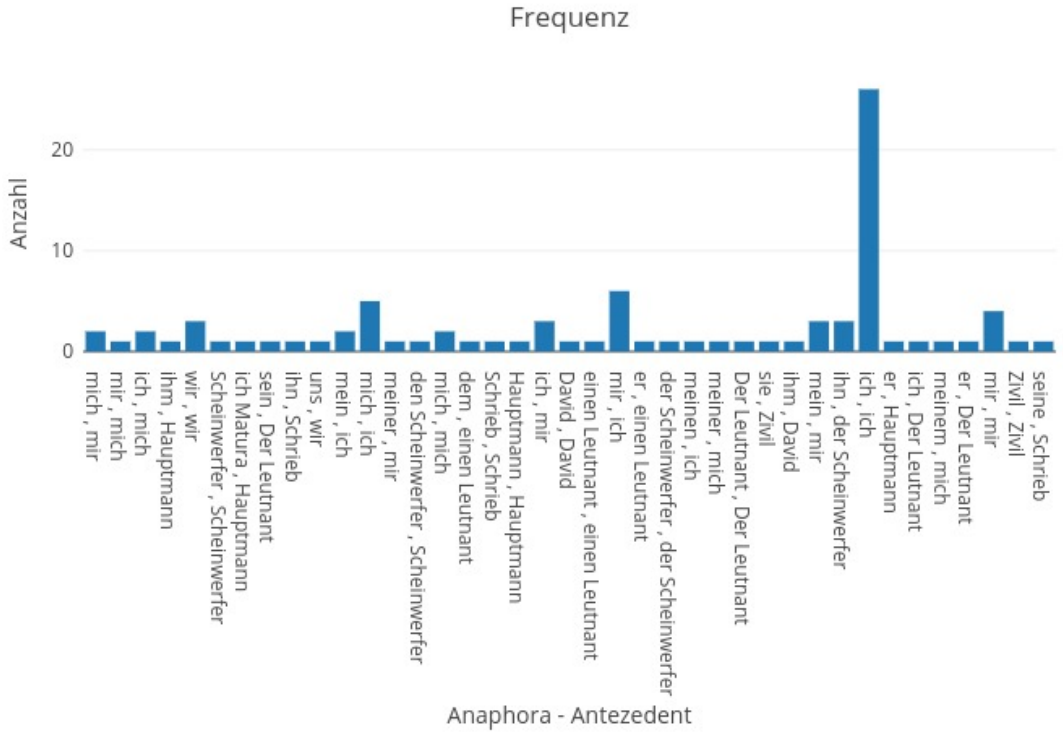

Abbildung 3.2: Frequenz der Koreferenz vor der Wortvektorenkorrektur

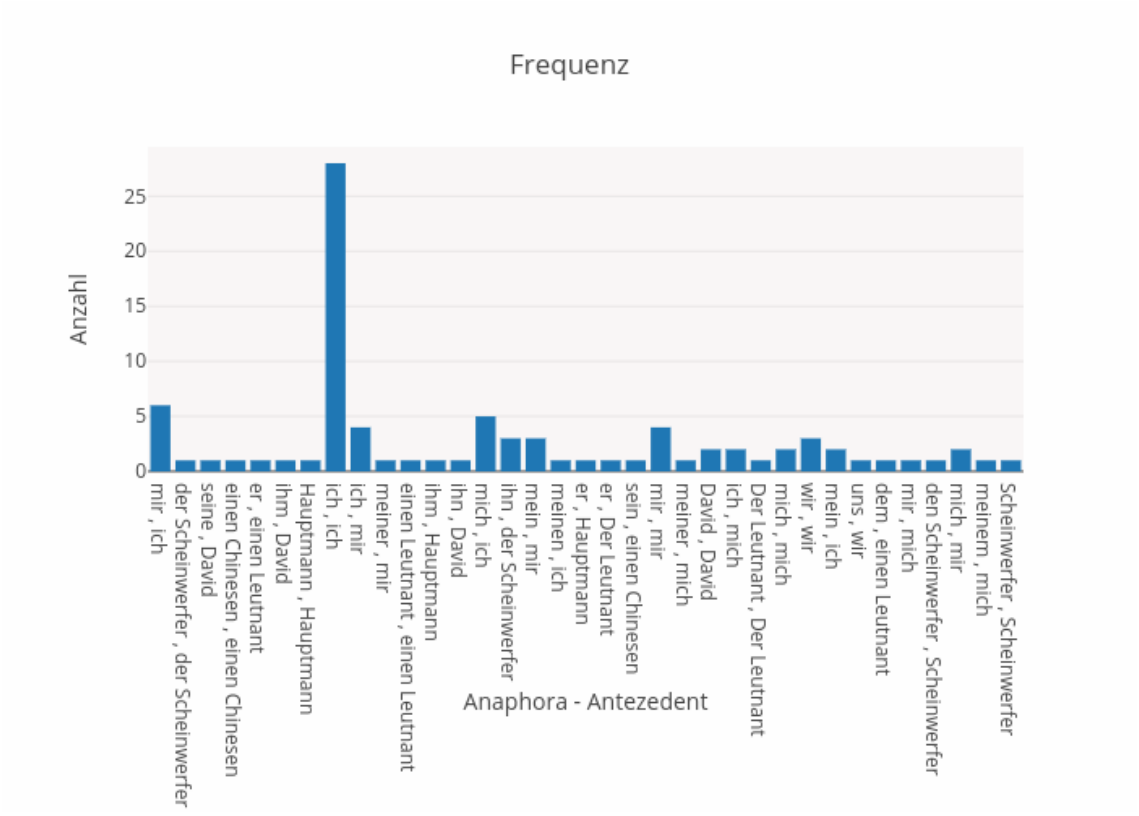

Abbildung 3.3: Frequenz der Koreferenz nach der Wortvektorenkorrektur

4 Merging der Manuskript XML-Datei mit erkannten Koreferenzen

In dem Kapitel „Merging der Manuskript XML-Datei mit erkannten Koreferenzen“ werden die notwendigen Schritte der Editionserweiterung genauer betrachtet. Anschließend werden die entstandenen Probleme erklärt und Lösungswege präsentiert.

4.1 Ausgangspunkt

```
−<s n="Ms-101,2r[2]_7" ana="facs:Ms-101,2r abnr:5 satznr:27">
   <lb/>
   von ihm um das Gefühl des Kontakts mit
   <lb/>
   meinem früheren Leben nicht zu verlieren.
   <PersName id="0" key="mich">Sehne</PersName>
 −<PersName id="1" key="mich">
     <PersName id="0" key="mich">mich</PersName>
   </PersName>
   <PersName id="2" key="mich">schon</PersName>
   <PersName id="3" key="mich">nach</PersName>
   <PersName id="4" key="mich">einem</PersName>
   <PersName id="5" key="mich">Brief</PersName>
 </s>
```

Abbildung 4.1: Falsche Zusammenführung

Bei der Implementierung wurde ein von Azada Rustamova entwickeltes Tool benutzt (siehe Anhang 1). Ihr Projektbericht ist auch als Anhang 2 beigelegt. Sie hat es für ein gemeinsames Projekt des CIS mit Historikern entwickelt, um die Suchmaschine für Leopold Wilhelms Briefe zu optimieren. Das entwickelte Skript benötigt Weiterentwicklungen und Optimierungen, um an Wittgensteins Daten angepasst zu werden. Zusätzlich lag der Schwerpunkt der Bachelorarbeit von Azar [2017] auf der Optimierung der linguistischen Suche des XML-annotierten Nachlasses von Ludwig Wittgenstein. Diese Arbeit diente als Hilfe für weitere, von mir durchführte, Untersuchungen.

Der Ausgangspunkt ist folgender: Obwohl das Tool im Einsatz der Historiker reibungslos funktioniert, muss es auf die Bedürfnisse des Manuskriptes angepasst werden.

Beispielsweise beim ersten Durchlauf werden die Positionen der Wörter vertauscht (siehe Abbildung 4.1). Alle Anaphoren bekommen als Referenz `mich`, obwohl es nur die Anaphoren `mich` und `meinen` betreffen sollte. Das liegt daran, dass die gesamten Sätze XML-Elemente bilden (siehe Abbildung 4.2).

In den Bemerkungen wurden die Tags `</sp>`, welche nach jedem `<w>`–Knoten vorkamen, gelöscht, damit die Zusammenführung der Dateien ermöglicht wird. Das dazu dienende Shell-Skript ist als CD beigelegt. Die einzelnen Wörter bilden nun XML-Elemente, welche die Attribute `Wortart` und `Lemma` beinhalten.

Wenn die einzelnen Wörter XML-Elemente bilden, wird die Perfomanz des Programms `merge_POS_coref.py` deutlich besser. Der folgende Abschnitt zeigt die neue Darstellung der Daten:

```
<s n="Ms-101,2r[2]_7" ana="facs:Ms-101,2r abnr:5 satznr:27">
<w t="VVFIN" l="sehnen">Sehne</w>
<w t="PPER" l="ich">mich</w>
<w t="ADV" l="schon">schon</w>
<w t="APPR" l="nach">nach</w>
<w t="ART" l="eine">einem</w>
<w t="NN" l="Brief">Brief</w>
<w t="APPR" l="von">von</w>
<w t="PPER" l="er">ihm</w>
</s>
```

```
<xml>
        <ab n="Ms-101,2r[2]" ana="abnr:5">
            <s n="Ms-101,2r[2]_1" ana="facs:Ms-101,2r abnr:5 satznr:21">Schlecht geschlafen (Ungeziefer).</s>
            <s n="Ms-101,2r[2]_2" ana="facs:Ms-101,2r abnr:5 satznr:22">Nachdem ich das Zimmer <lb/>gekehrt hatte marschierten wir zu ein paar
  alten Mörsern <lb/> &amp;amp; wurden im Gebrauch instruiert.</s>
            <s n="Ms-101,2r[2]_3" ana="facs:Ms-101,2r abnr:5 satznr:23">Furchtbar heiß.</s>
            <lb/>
            <s n="Ms-101,2r[2]_4" ana="facs:Ms-101,2r abnr:5 satznr:24">Das Essen ist uneßbar.</s>
            <s n="Ms-101,2r[2]_5" ana="facs:Ms-101,2r abnr:5 satznr:25">Werde vielleicht in Zukunft <lb/>außerhalb der Kaserne schlafen.</s>
            <s n="Ms-101,2r[2]_6" ana="facs:Ms-101,2r abnr:5 satznr:26">An <persName key="Pinsent, David Hume">David</persName>
               <lb/> geschrieben.</s>
            <s n="Ms-101,2r[2]_7" ana="facs:Ms-101,2r abnr:5 satznr:27">Sehne mich schon nach einem Brief <lb/>von ihm um das Gefühl des Kontakts
  mit <lb/>meinem früheren Leben nicht zu verlieren.</s>
            <s n="Ms-101,2r[2]_8" ana="facs:Ms-101,2r abnr:5 satznr:28">Noch <lb/>nicht gearbeitet.</s>
         </ab>

 </xml>
```

Abbildung 4.2: Sätze als XML-ELemente in der Basisdatei

4.2 Vorgehensweise

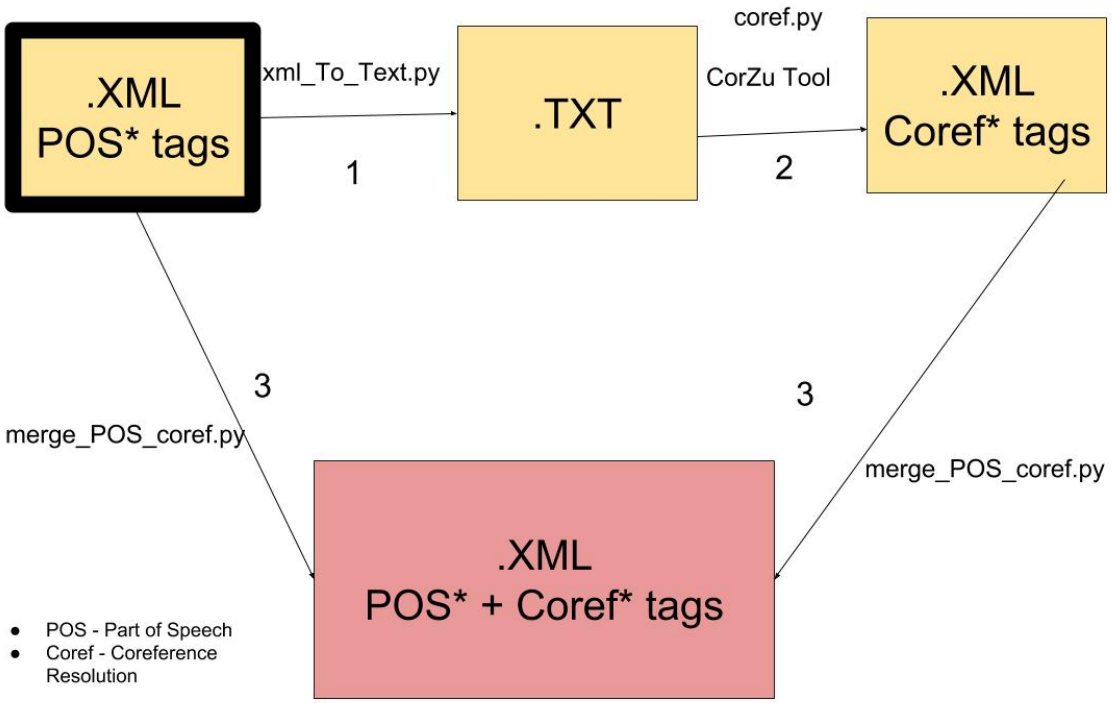

Abbildung 4.3: Vorgehensweise

Abbildung 4.3 zeigt wie der Versuch aufgebaut wurde. Als Erstes wurden die Bemerkungen verwendet, in denen die einzelnen Wörter XML-Elemente bilden. Des Weiteren

haben sie auch die Attribute `Wortart` und `Lemma`. Diese Form passt besser zur Erweiterung der XML-Edition des Nachlasses. Sie wird abschließend als Basisdatei für die Zusammenführung der Koreferenzannotationen dienen. Als Nächstes werden mit Hilfe des Skripts `xml_to_Text.py` die Bemerkungen in ein Text-Format konvertiert. Das Tool CorZu benötigt hierbei eine Texteingabe. Das Skript `coref.py` ruft das Tool auf. Die Ausgabe erfolgt im XML-Format. Die Koreferenzannotationen veranschaulichen die Reihenfolge der Antezedenten in der Koreferenzkette und der Antezedenten selbst. Die Reihenfolge wird durch das Attribut `id` festgelegt. Der Antezedent wird durch das Attribut `key` markiert. Abschließend führt das Skript `merge_POS_coref.py` zwei XML-Dateien zusammen.

```
1. Aufruf: sh nosp.sh
Input: ~/koreferenzen/lib/dev/xml/*xml
Output: ~/koreferenzen/lib/dev/xml/nosp/*.new.xml

2. Aufruf: python3 xml_to_text.py original.xml output.txt
~/koreferenzen/lib$ python3 xml_to_text.py dev/xml/nosp/2r[3].xml.new.xml//
dev/xml/text/2r[3].txt

Input: dev/xml/nosp/*.xml
Output: dev/xml/text/*.txt

3. Aufruf: python3 coref.py input.txt coref.xml
~/koreferenzen/ext$ python3 coref.py ~/koreferenzen/lib/dev/xml/text/2r[3].txt//
~/koreferenzen/lib/dev/coref/2r[3]c.xml
Input: ~/koreferenzen/lib/dev/xml/text/*.txt
Output: ~/koreferenzen/lib/dev/coref/*.xml

4. Aufruf: python3 merge_POS_coref.py original.xml coref.xml merged.xml
~/koreferenzen/lib$ python3 merge_POS_coref.py dev/xml/nosp/21v[2].xml.new.xml//
dev/coref/21v[2]c.xml dev/merged/21v[2]m.xml
'''
Die manuell korrigierten Koreferenzdateien sind 2r_3cor.xml, 5r_3cor.xml,//
7r_3cor.xml.
'''
Input1: dev/xml/*xml
Input2:dev/coref/*.xml
Output: dev/merged/*.xml
```

4.3 Struktur der Knoten

Im folgenden Abschnitt wird auf die Struktur der einfachen Knoten von XML-Dateien eingegangen. Folgende Darstellung der Knoten erklärt wie die Funktionen im Skript `merge_POS_coref.py` aufgebaut sind.

Die Abbildung 4.4 schildert die Struktur eines Knotens der wortartgetaggten XML-Datei. Das Element ist `w`. Das Attribut `t` entspricht dem `PPER` (Personalpronomen). Das Attribut `l` (Lemma) wiederum entspricht dem Pronomen `ich`. Hier wurde das Verfahren des Wortartentaggers übernommen, indem Wortartentags und Lemmas definiert wurden. Das Wort `mir` im unteren rechten Kreis stellt den Taginhalt dar. Die Abbildung 4.5 beleuchtet die Struktur eines Knotens der XML-Datei mit Koreferenzannotation. Das Element heißt `coref`. Das Attribut `id` umfasst die Reihenfolge des Antezedenten im Text, der in eine Beziehung mit der Anaphora gesetzt wurde. Das Attribut `key` steht für den Antezedenten `ich`. Die Anaphora bzw. der Annotationsinhalt `mir`, ist im rechten Kreis dargestellt. Die

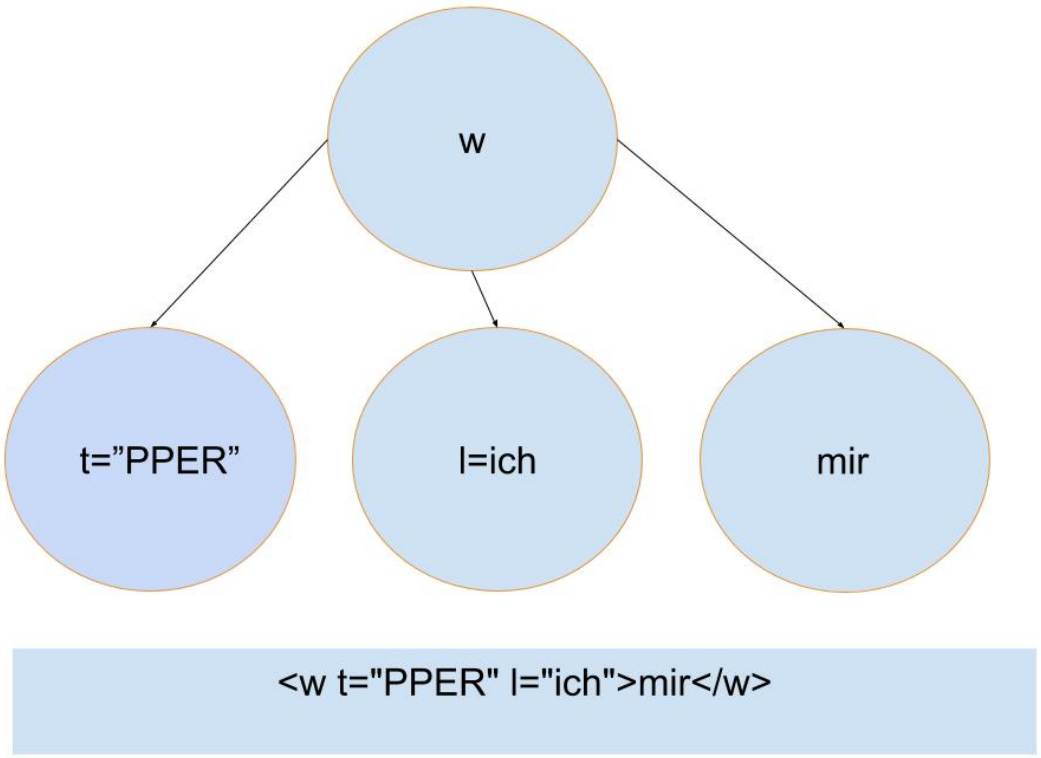

Abbildung 4.4: 1 Knoten in der Basisdatei

Abbildung 4.6 erklärt, wie die Zusammenführung durchgeführt wird. Hiermit wird das Element `w` modifiziert. Der Taginhalt des Elementes `w`, der aus dem Wort `mir` besteht, wird über die Liste von Taginhalten der Elemente `<coref>` gesucht. Falls sie übereinstimmen, wird der Taginhalt des Elementes `w` gelöscht und an seiner Stelle ein Tochterknoten hinzugefügt. Der Tochterknoten ist das Element `coref`. Da das Attribut `id` für die Suchmaschine WITTFind nicht relevant ist, wird es nicht in der zusammengeführten Datei vorhanden sein. Das Attribut `<isMult>` wird auf `0` gesetzt, falls der Taginhalt eines entsprechenden `<coref>`–Knotens aus einem Wort besteht. Wenn der Taginhalt des Knotens aus dem Multiple String besteht, wird das Attribut `<isMult>` weiter nummeriert.

Wenn das Attribut `key` einen Wert aus der folgenden Liste hat, wird es automatisch mit dem Wert `Ludwig Wittgenstein` umgetauscht.

```
['ich','mir','mich']
```

Beispiel:

```
1.
<w t="PPER" l="ich"><coref key="Ludwig Wittgenstein" isMult="0">ich</coref></w>
2.
<w t="PRF" l="ich"><coref key="Ludwig Wittgenstein" isMult="0">mich</coref></w>
3.
<w t="PPER" l="ich"><coref key="Ludwig Wittgenstein" isMult="0">mir</coref></w>
4.
<w t="PPER" l="ich"><coref key="Ludwig Wittgenstein" isMult="0">ich</coref></w>
```

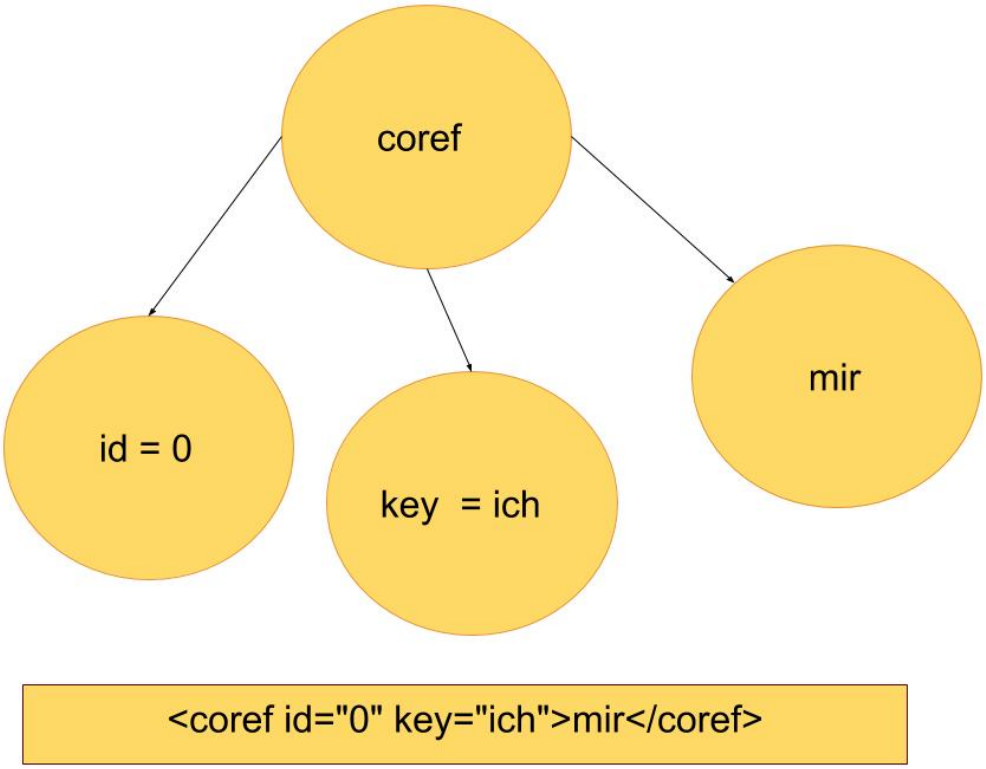

Abbildung 4.5: 1 Knoten in der Koreferenzdatei

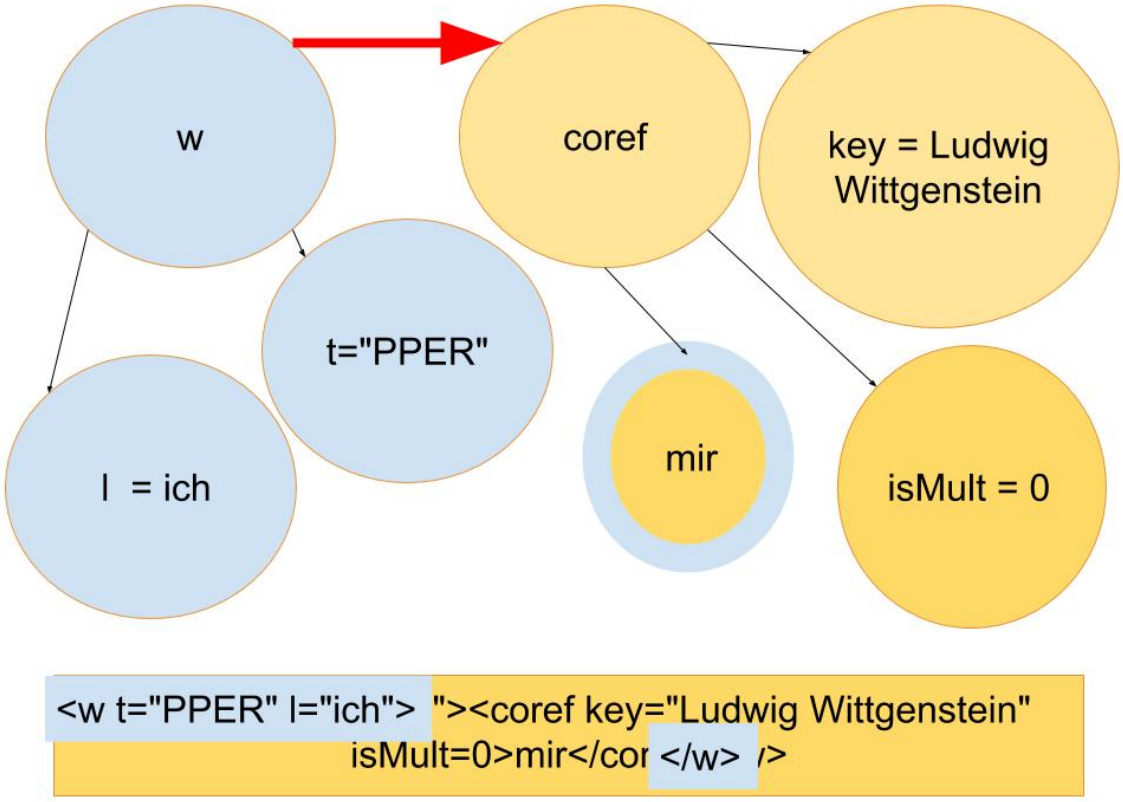

Abbildung 4.6: 1 Knoten nach dem Merging

4.4 Herausforderungen bei der Aufgabenstellung

4.4.1 Multiple String unter dem Attribut key

Das Problem ist, dass nach der Koreferenzauflösung, die Koreferenten als Attribut einen Antezedenten haben, welcher 2 Wörter als Value enthält. Um die Werte weiterzubearbeiten, wurde die Funktion `extract_node` angepasst. Diese Funktion kann auch Werte der Attribute mit multiple String abfragen. Die folgende Beispiele stellen den Fall mit multiple String als Wert des Attributes `key` dar.

```
1.
<w t="ART" l="eine"><coref key="einen Leutnant" isMult="0">einen</coref></w>
2.
<w t="NN" l="Leutnant"><coref key="einen Leutnant" isMult="1">Leutnant//
</coref></w>
3.
<w t="PRELS" l="die"><coref key="einen Leutnant" isMult="0">dem</coref></w>
4.
<w t="PPER" l="er"><coref key="einen Leutnant" isMult="0">er</coref></w>
5.
<w t="PPER" l="er"><coref key="Der Leutnant" isMult="0">Er</coref></w>
6.
<w t="PPER" l="er"><coref key="der Scheinwerfer" isMult="0">ihn</coref></w>
7.
<w t="VAINF" l="sein"><coref key="einen Chinesen" isMult="0">sein</coref></w>
```

4.4.2 Multiple String als Taginhalt

In der ursprünglichen Datei kommen einzelne Wörter als Knoten vor. Die Abbildung 4.7 illustriert den Fall. Das Wort `der` hat die Wortart `ART` (Artikel) und das Lemma `die`. Auf die gleiche Art und Weise ist das Wort `Scheinwerfer` annotiert. Die Wortart ist `NN` (Nomen) und das Lemma `Scheinwerfer`.

Im Gegensatz zur Abbildung 4.7 kommen in der koreferenzannotierten Datei der Abbildung 4.8 mehrere Wörter als Inhalt in einem Knoten vor (siehe rechter Kreis `der Scheinwerfer`). Anschließend wurde die Zusammenführung der Elemente, welche in der wortartgetaggten Datei einzelne Elemente präsentieren, aber nach der Koreferenzauflösung als Inhalt in einem Knoten vorkommen, optimiert. Die Abbildung 4.9 schildert dies. Die Elemente in der wortartgetaggten Datei werden mit dem Attribut `<isMult>` durchnummeriert, welches zeigt, in welcher Reihenfolge das Wort im Koreferenzknoteninhalt vorkam. Die Struktur der wortartgetaggten XML-Datei wurde beibehalten. Zur Vereinfachung werden lediglich jeweils 2 Nachbarknoten nach dem Merging dargestellt, dennoch ist das Ergebnis eindeutig.

```
1.
<w t="ART" l="die"><coref key="Der Leutnant" isMult="0">Der</coref></w>
<w t="NN" l="Leutnant">//
<coref key="Der Leutnant" isMult="1">Leutnant</coref></w>
2
<w t="ART" l="die"><coref key="der Scheinwerfer" isMult="0">der</coref></w>
<w t="NN" l="Scheinwerfer">//
<coref key="der Scheinwerfer" isMult="1">Scheinwerfer</coref></w>
3.
<w t="ART" l="eine"><coref key="einen Chinesen" isMult="0">einen</coref></w>
<w t="NN" l="Chinese"><coref key="einen Chinesen" isMult="1">Chinesen</coref></w>
```

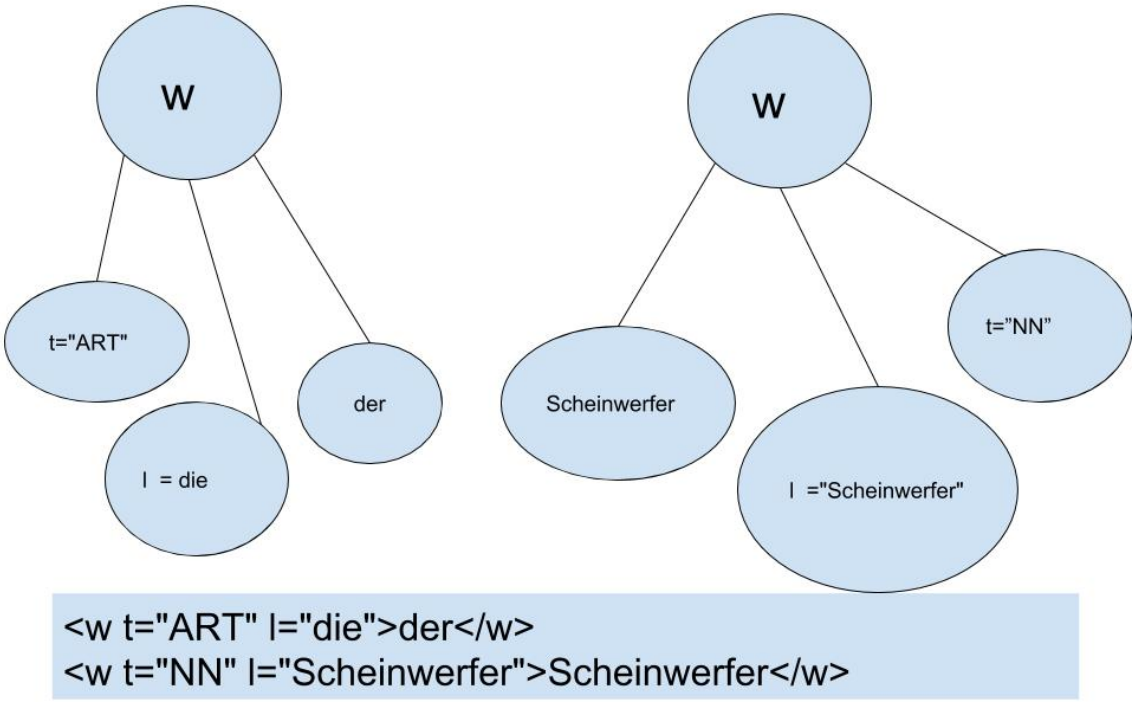

Abbildung 4.7: 2 Knoten in der Basisdatei

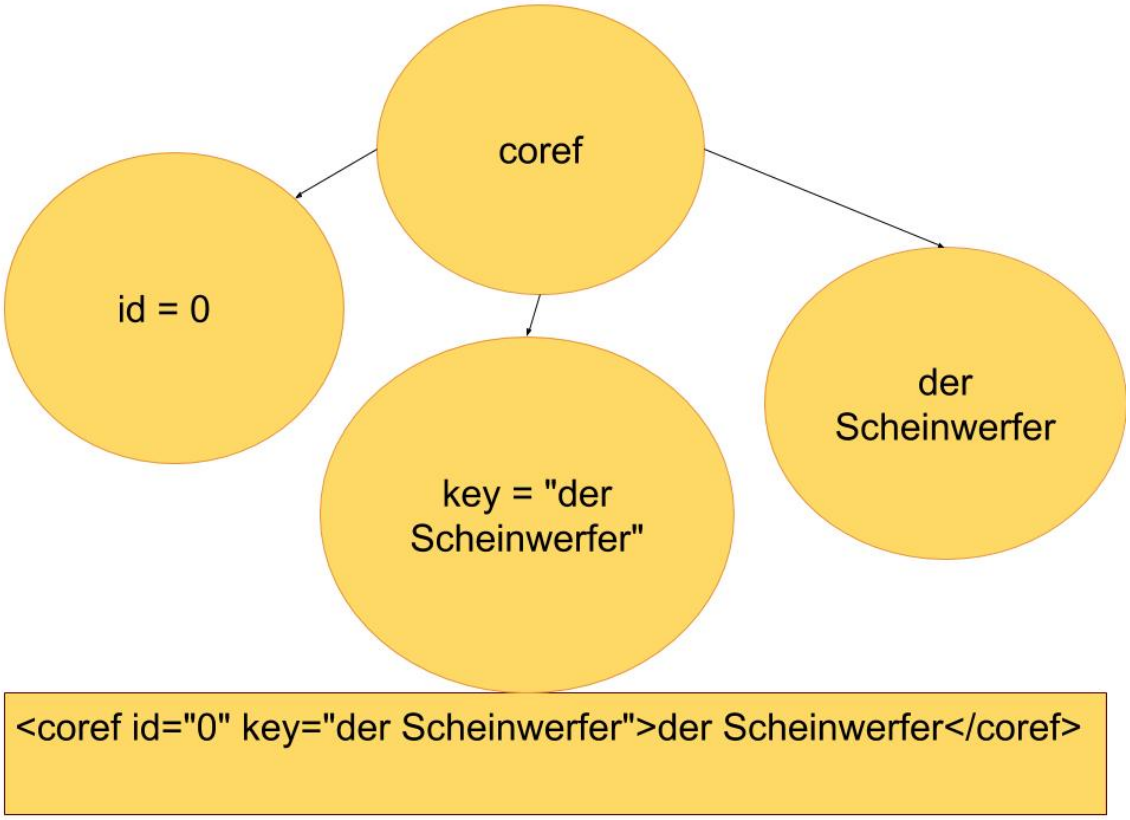

Abbildung 4.8: 1 Knoten mit multiple String

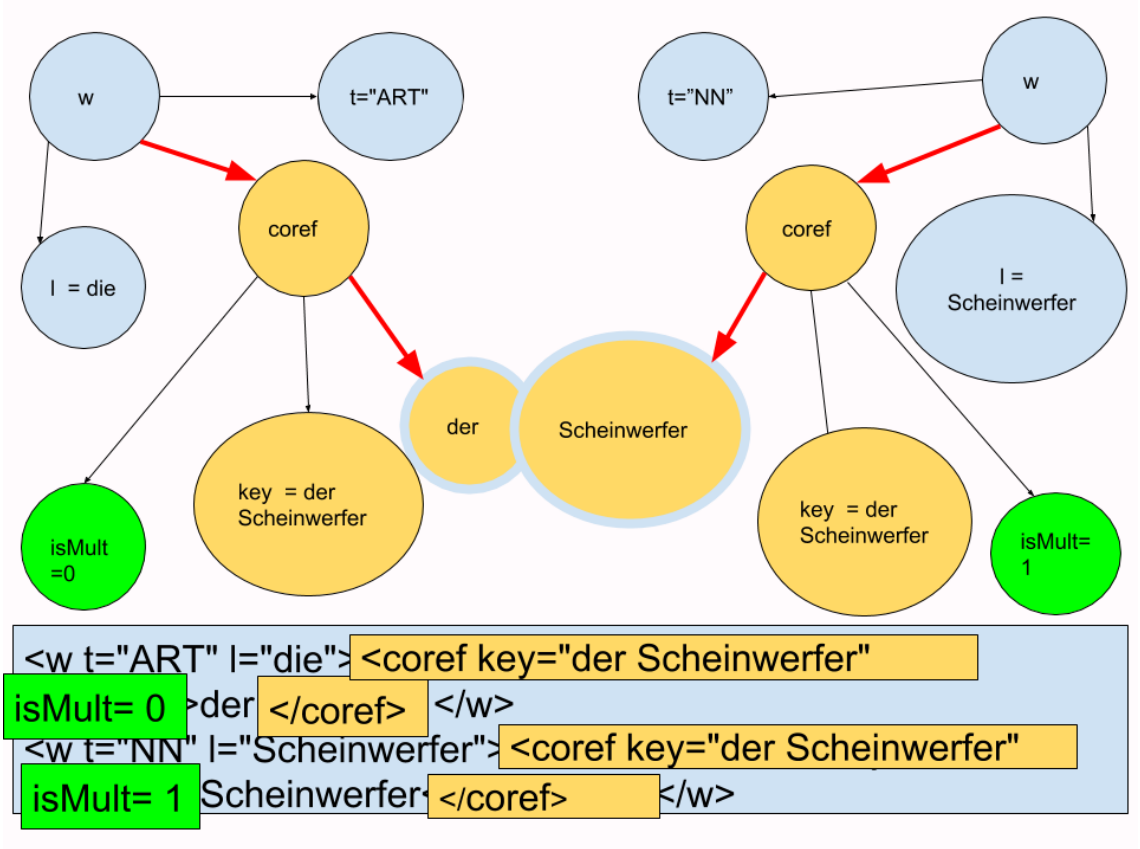

Abbildung 4.9: 2 Knoten nach dem Merging

5 Evaluierung

Um die Umsetzung der Koreferenzauflösung zu bewerten, wurde die Anwendung auf Personennamen beschränkt. Als Grundlage für die Evaluierung diente eine angefertigte Liste mit Personennamen.

```
[Augustinus, Dedekind, Frege, Russell]
```

Es wurde auch ein Pseudosatz erstellt, um die Perfomanz der Koreferenzauflösung überprüfen zu können. Es handelt sich um 2 Personennamen und mehreren Pronomen in einem Satz. Anschließend wurden die ausgewählten Textabschnitte vorbereitet, indem mit Hilfe eines Shell-Skripts die unnötigen Tags gelöscht wurden. Das zur Anwendung kommende Skript lautete folgendermaßen:

```
#!/bin/bash
cd ~/eval
mkdir nosp
for i in *.xml;
    do sed -E 's/<sp\/>//g' $i > nosp/$i.new.xml;
done
```

5.1 Russell in Ms-101,33v[2]_7, Ms-101,33v[2]_8

Der zweite Satz enthält die Anaphora `er`, welche sich auf den Namen `Russell` bezieht. Das erwartete Ergebnis nach der Zusammenführung der gezeigten wortartgetaggten Datei mit der koreferenzaufgelösten Datei soll den Tag `<coref>` beinhalten. Das Attribut `keyPersName` zeigt den Antezedenten `Russell`.

Wortartgetaggte Datei

```
<s>
<s n="Ms-101,33v[2]_7" ana="facs:Ms-101,33v abnr:44 satznr:300">
<w t="VVFIN" l="denken">Dachte</w>
<w t="APPR" l="in">in</w>
<w t="ART" l="die">den</w>
<w t="ADJA" l="letzt">letzten</w>
<w t="NN" l="Tag">Tagen</w>
<w t="ADV" l="oft">oft</w>
<w t="APPR" l="an">an</w>
<persName key="Russell, Bertrand">
<w t="NE" l="Russell">Russell</w></persName>
<w t="$." l=".">.</w>
</s>
<s n="Ms-101,33v[2]_8" ana="facs:Ms-101,33v abnr:44 satznr:301">
<w t="KOUS" l="ob">Ob</w>
<w t="PPER" l="er">er</w>
<w t="ADV" l="noch">noch</w>
<w t="APPR" l="an">an</w>
<w t="PPER" l="ich">mich</w>
<w t="VVFIN" l="denken">denkt</w>
```

```
<w t="$." l="?">?</w>
</s>
</s>
```

Koreferenzaufgelöste Datei

Als Erstes wird die XML Datei in ein Textformat konvertiert, um die Koreferenzen durch das Tool CorZu festzustellen. Das Tool wiederum erstellt eine XML-Datei mit Koreferenzmarkierungen.

```
<Coref> Dachte in den letzten Tagen
oft an <coref id="0" key="Russell">Russell</coref>.
Ob <coref id="0" key="Russell">er</coref> noch an mich denkt?
</Coref>
```

Zusammengeführte Datei

Um die Zusammenführung zu ermöglichen, werden die beschriebenen Schritte in den Kapiteln 3 und 4 ausgeführt. Anbei wird die zusammengeführte Datei veranschaulicht.

```
<?xml version='1.0' encoding='ASCII'?>
<s>
<s n="Ms-101,33v[2]_7" ana="facs:Ms-101,33v abnr:44 satznr:300">
<w t="VVFIN" l="denken">Dachte</w>
<w t="APPR" l="in">in</w>
<w t="ART" l="die">den</w>
<w t="ADJA" l="letzt">letzten</w>
<w t="NN" l="Tag">Tagen</w>
<w t="ADV" l="oft">oft</w>
<w t="APPR" l="an">an</w>
<persName key="Russell, Bertrand">
<w t="NE" l="Russell"><coref key="Russell" /
keyPersName="Russell" isMult="0">Russell</coref></w>
</persName>
<w t="$." l=".">.</w>
</s>
<s n="Ms-101,33v[2]_8" ana="facs:Ms-101,33v abnr:44 satznr:301">
<w t="KOUS" l="ob">Ob</w>
<w t="PPER" l="er"><coref key="Russell" /
keyPersName="Russell" isMult="0">er</coref></w>
<w t="ADV" l="noch">noch</w>
<w t="APPR" l="an">an</w>
<w t="PPER" l="ich">mich</w>
<w t="VVFIN" l="denken">denkt</w>
<w t="$." l="?">?</w>
</s>
</s>
```

Kurzfassung

Dem Pronomen `er` wurde der Antezedent `Russell` zugewiesen. Die Annotation erweitert die Edition des Wittgenstein Nachlasses.

```
/* vor der Erweiterung der Edition */
<w t="PPER" l="er">er</w>
```

```
/* nach der Erweiterung der Edition */
<w t="PPER" l="er"><coref key="Russell" /
keyPersName="Russell" isMult="0">er</coref></w>
```

5.2 Frege in Ts-213,3r[2]_1

Hier wird ein Satz mit dem Eigennamen Frege untersucht.

Wortartgetaggte Datei

```
<s n="Ts-213,3r[2]_1" ana="facs:Ts-213,3r /
abnr:176 satznr:412">
<w t="KOUS" l="wenn">Wenn</w>
<persName key="Frege, Gottlob">
<w t="NE" l="Frege">Frege</w>
</persName>
<w t="APPR" l="gegen">gegen</w>
<w t="ART" l="die">die</w>
<w t="ADJA" l="formal">formale</w>
<w t="NN" l="Auffassung">Auffassung</w>
<w t="ART" l="die">der</w>
<w t="NN" l="Arithmetik">Arithmetik</w>
<w t="VVFIN" l="sprechen">spricht</w>
<w t="$," l=",">,</w>
<w t="ADV" l="so">so</w>
<w t="VVFIN" l="sagen">sagt</w>
<w t="PPER" l="er">er</w>
<w t="ADJD" l="gleichsam">gleichsam</w>
<w t="$." l=":">:</w>
<w t="PDAT" l="dies">diese</w>
<w t="ADJA" l="kleinlich">kleinlichen</w>
<w t="NN" l="Erklaerung">Erklaerungen</w>
<w t="$," l=",">,</w>
<w t="ART" l="die">die</w>
<w t="NN" l="Symbol">Symbole</w>
<w t="ADJD" l="betreffend">betreffend</w>
<w t="$," l=",">,</w>
<w t="VAFIN" l="sein">sind</w>
<w t="ADJD" l="muessig">muessig</w>
<w t="$," l=",">,</w>
<w t="KOUS" l="wenn">wenn</w>
<w t="PPER" l="wir">wir</w>
<w t="PDS" l="diese">diese</w>
<w t="VVFIN" l="verstehen">verstehen</w>
<w t="$." l=".">.</w>
</s>
```

Koreferenzaufgelöste Datei

Das Pronomen `er` bezieht sich auf den Eigennamen. Zusätzlich dazu gibt es noch ein Koreferenzpaar `die Symbole - diese`, welches auch von dem Tool CorZu erkannt wurde.

```
<Coref>/
Wenn <coref id="0" key="Frege">Frege</coref> gegen /
```

```
die formale Auffassung der Arithmetik spricht, /
so sagt <coref id="0" key="Frege">er</coref> gleichsam: /
diese kleinlichen Erklaerungen, /
<coref id="1" key="die Symbole">die Symbole</coref> betreffend, /
sind muessig, wenn wir <coref id="1" /
key="die Symbole">diese</coref> verstehen./
</Coref>
```

Zusammengeführte Datei

Die Anaphora `er` wird nach der Zusammenführung der Dateien mit dem Attribut `keyPersName` markiert.

`die Symbole` ist ein multiple String. Das Verfahren, welches beschreibt, wie ein multiple String in der Korefernzdatei mit Wittgensteins Basisdatei zusammengeführt wird und welche Herausforderungen es darstellt, wurde bereits in Kapitel 4 beschrieben. Der multiple String `die Symbole` wird mit Hilfe des Attributs `isMult = 0` als erstes Wort im String, bzw. `isMult = 1` als zweites Wort hervorgehoben.

```
<?xml version='1.0' encoding='ASCII'?>
<s n="Ts-213,3r[2]_1" ana="facs:Ts-213,3r /
abnr:176 satznr:412">
<w t="KOUS" l="wenn">Wenn</w>
<persName key="Frege, Gottlob">
<w t="NE" l="Frege"><coref key="Frege" keyPersName="Frege" /
isMult="0">Frege</coref></w>
</persName>
<w t="APPR" l="gegen">gegen</w>
<w t="ART" l="die">die</w>
<w t="ADJA" l="formal">formale</w>
<w t="NN" l="Auffassung">Auffassung</w>
<w t="ART" l="die">der</w>
<w t="NN" l="Arithmetik">Arithmetik</w>
<w t="VVFIN" l="sprechen">spricht</w>
<w t="$," l=",">,</w>
<w t="ADV" l="so">so</w>
<w t="VVFIN" l="sagen">sagt</w>
<w t="PPER" l="er"><coref key="Frege" keyPersName="Frege" /
isMult="0">er</coref></w>
<w t="ADJD" l="gleichsam">gleichsam</w>
<w t="$." l=":">:</w>
<w t="PDAT" l="dies"><coref key="die Symbole" /
isMult="0">diese</coref></w>
<w t="ADJA" l="kleinlich">kleinlichen</w>
<w t="NN" l="Erklaerung">Erklaerungen</w>
<w t="$," l=",">,</w>
<w t="ART" l="die"><coref key="die Symbole" /
isMult="0">die</coref></w>
<w t="NN" l="Symbol"><coref key="die Symbole" /
isMult="1">Symbole</coref></w>
<w t="ADJD" l="betreffend">betreffend</w>
<w t="$," l=",">,</w>
<w t="VAFIN" l="sein">sind</w>
<w t="ADJD" l="muessig">muessig</w>
<w t="$," l=",">,</w>
```

```
<w t="KOUS" l="wenn">wenn</w>
<w t="PPER" l="wir">wir</w>
<w t="PDS" l="diese"><coref key="die Symbole" /
isMult="0">diese</coref></w>
<w t="VVFIN" l="verstehen">verstehen</w>
<w t="$." l=".">.</w>
</s>
```

Kurzfassung

Die Evaluierung beschränkt sich auf Personennamen. Im Satz Ts-213,3r[2]_1 wurde die Koreferenz `Frege - er` erfolgreich umgesetzt.

```
/*vor der Erweiterung */
isMult="0">er</coref></w>

/*nach der Erweiterung */
<w t="PPER" l="er"><coref key="Frege" keyPersName="Frege" /
isMult="0">er</coref></w>
```

5.3 Dedekind in Ts-213,742r[3]et743r[1]_5

In diesem Abschnitt wird gezeigt, wie ein Antezedent, in Form des Personennamens `Dedekind`, und eine Anaphora `er` gefunden wurden und anschließend die Edition beim Merging der Dateien mit dem Attribut `keyPersName` erweitert wird.

Wortartgetaggte Datei

```
<s n="Ts-213,742r[3]et743r[1]_5" ana="facs:Ts-213,742r /
abnr:2701 satznr:9297">
<w t="ADV" l="so">So</w>
<w t="VVFIN" l="versuchen">versucht</w>
<persName key="Dedekind, Richard">
<w t="NE" l="Dedekind">Dedekind</w>
</persName>
<w t="ART" l="eine">eine</w>
<w t="ADJA" l="unendlich">unendliche</w>
<w t="NN" l="Klasse">Klasse</w>
<w t="PTKZU" l="zu">zu</w>
<emph rend="space">
<w t="VVINF" l="beschreiben">beschreiben</w>
</emph>
<w t="$." l=";">;</w>
<w t="KOUS" l="indem">indem</w>
<w t="PPER" l="er">er</w>
<w t="VVFIN" l="sagen">sagt</w>
<w t="$," l=",">,</w>
<w t="PPER" l="es">es</w>
<w t="VAFIN" l="sein">sei</w>
<w t="PIS" l="eine">eine</w>
<w t="$," l=",">,</w>
<w t="PRELS" l="die">die</w>
<w t="ART" l="eine">einer</w>
<w t="ADJA" l="echt">echten</w>
```

```
<w t="NN" l="Teilklasse">Teilklasse</w>
<w t="PPOSAT" l="ihr">ihrer</w>
<w t="ADV" l="selbst">selbst</w>
<w t="ADJD" l="aehnlich">aehnlich</w>
<w t="VAFIN" l="sein">ist</w>
<w t="$." l=".">.</w>
</s>
```

Koreferenzaufgelöste Datei

```
<Coref> /
So versucht <coref id="0" key="Dedekind">Dedekind</coref> /
<coref id="1" key="eine unendliche Klasse">eine unendliche Klasse</coref> /
zu beschreiben; indem <coref id="0" key="Dedekind">er</coref> sagt, /
es sei eine, <coref id="1" key="eine unendliche Klasse">die</coref> /
einer echten Teilklasse /
<coref id="1" key="eine unendliche Klasse">ihrer</coref>/
selbst aehnlich ist./
</Coref>
```

Zusammengeführte Datei

Die Koreferenz `Dedekind - er` wurde vom Tool CorZu annotiert und von dem Python-Skript mit dem Attribut `keyPersName` markiert.

```
<?xml version='1.0' encoding='ASCII'?>
<s n="Ts-213,742r[3]et743r[1]_5" ana="facs:Ts-213,742r abnr:2701 satznr:9297">
<w t="ADV" l="so">So</w>
<w t="VVFIN" l="versuchen">versucht</w>
<persName key="Dedekind, Richard">
<w t="NE" l="Dedekind"><coref key="Dedekind" /
keyPersName="Dedekind" isMult="0">Dedekind</coref></w>
</persName>
<w t="ART" l="eine">eine</w>
<w t="ADJA" l="unendlich">unendliche</w>
<w t="NN" l="Klasse">Klasse</w>
<w t="PTKZU" l="zu">zu</w>
<emph rend="space">
<w t="VVINF" l="beschreiben">beschreiben</w>
</emph>
<w t="$." l=";">;</w>
<w t="KOUS" l="indem">indem</w>
<w t="PPER" l="er"><coref key="Dedekind" keyPersName="Dedekind" /
isMult="0">er</coref></w>
<w t="VVFIN" l="sagen">sagt</w>
<w t="$," l=",">,</w>
<w t="PPER" l="es">es</w>
<w t="VAFIN" l="sein">sei</w>
<w t="PIS" l="eine">eine</w>
<w t="$," l=",">,</w>
<w t="PRELS" l="die"><coref key="eine unendliche Klasse" /
isMult="0">die</coref></w>
<w t="ART" l="eine">einer</w>
<w t="ADJA" l="echt">echten</w>
<w t="NN" l="Teilklasse">Teilklasse</w>
```

```
<w t="PPOSAT" l="ihr"><coref /
key="eine unendliche Klasse" isMult="0">ihrer</coref></w>
<w t="ADV" l="selbst">selbst</w>
<w t="ADJD" l="aehnlich">aehnlich</w>
<w t="VAFIN" l="sein">ist</w>
<w t="$." l=".">.</w>
</s>
```

Kurzfassung

Im Satz Ts-213,742r[3]et743r[1]_5 wurde die Koreferenz `Dedekind - er` erfolgreich umgesetzt.

```
/*vor der Erweiterung */
<w t="PPER" l="er">er</w>

/*nach der Erweiterung */
<w t="PPER" l="er"><coref key="Dedekind" /
keyPersName="Dedekind" isMult="0">er</coref></w>
```

5.4 Augustinus

In zwei anschließenden Sätzen wurden die Koreferenzen `Augustinus - er` bei der Editionserweiterung berücksichtigt.

5.4.1 Augustinus in Ms-110,178[3]_1

Wortartgetaggte Datei

```
<s n="Ms-110,178[3]_1" ana="facs:Ms-110,178 abnr:943 /
satznr:1864">
<w t="ADV" l="so">So</w>
<w t="VAFIN" l="sein">war</w>
<w t="ADV" l="also">also</w>
<persName key="Augustinus, Aurelius">
<w t="NE" l="Augustinus">Augustinus</w>
</persName>
<w t="APPRART" l="in">im</w>
<w t="NN" l="Irrtum">Irrtum</w>
<w t="KOUS" l="wenn">wenn</w>
<w t="PPER" l="er">er</w>
<w t="NN" l="Gott">Gott</w>
<w t="APPR" l="auf">auf</w>
<w t="PIAT" l="jede">jeder</w>
<w t="NN" l="Seite">Seite</w>
<w t="ART" l="die">der</w>
<w t="NN" l="Confession">Confessionen</w>
<w t="VVFIN" l="anrufen">anruft</w>
<w t="$." l="?">?</w>
</s>
```

Koreferenzaufgelöste Datei

```
<Coref> So war also <coref id="0" key="Augustinus">Augustinus</coref>/
im Irrtum wenn <coref id="0" key="Augustinus">er</coref> /
Gott auf jeder Seite der Confessionen anruft?</Coref>
```

Zusammengeführte Datei

```
<?xml version='1.0' encoding='ASCII'?>
<s n="Ms-110,178[3]_1" ana="facs:Ms-110,178 abnr:943 satznr:1864">
<w t="ADV" l="so">So</w>
<w t="VAFIN" l="sein">war</w>
<w t="ADV" l="also">also</w>
<persName key="Augustinus, Aurelius">
<w t="NE" l="Augustinus"><coref key="Augustinus" /
keyPersName="Augustinus" isMult="0">Augustinus</coref></w>
</persName>
<w t="APPRART" l="in">im</w>
<w t="NN" l="Irrtum">Irrtum</w>
<w t="KOUS" l="wenn">wenn</w>
<w t="PPER" l="er"><coref key="Augustinus" /
keyPersName="Augustinus" isMult="0">er</coref></w>
<w t="NN" l="Gott">Gott</w>
<w t="APPR" l="auf">auf</w>
<w t="PIAT" l="jede">jeder</w>
<w t="NN" l="Seite">Seite</w>
<w t="ART" l="die">der</w>
<w t="NN" l="Confession">Confessionen</w>
<w t="VVFIN" l="anrufen">anruft</w>
<w t="$." l="?">?</w>
</s>
```

Kurzfassung

Im Satz Ms-110,178[3]_1 wurde die Koreferenz `Augustinus - er` erfolgreich umgesetzt.

```
/*vor der Erweiterung */
<w t="PPER" l="er">er</w>

/* nach der Erweiterung*/
<w t="PPER" l="er"><coref key="Augustinus" /
keyPersName="Augustinus" isMult="0">er</coref></w>
```

5.4.2 Augustinus in Ts-213,26r[3]_4.1

Wortartgetaggte Datei

```
<s n="Ts-213,26r[3]_4.1" ana="facs:Ts-213,26r abnr:277 satznr:741">
<w t="PIS" l="man">Man</w>
<w t="VMFIN" l="koennen">koennte</w>
<w t="ADV" l="also">also</w>
<w t="VVINF" l="sagen">sagen</w>
<w t="$," l=",">,</w>
<persName key="Augustinus, Aurelius">
<w t="NE" l="Augustinus">Augustinus</w>
</persName>
<choice type="s">
<seg n="s_alt1">
<w t="VVFIN" l="stellen">stelle</w>
<w t="ART" l="die">das</w>
<w t="NN" l="Lernen">Lernen</w>
<w t="ART" l="die">der</w>
```

```
<w t="NN" l="Sprache">Sprache</w>
<w t="PTKA" l="zu">zu</w>
<w t="ADJD" l="einfach">einfach</w>
<w t="PTKVZ" l="dar">dar</w>
</seg>
</choice>
<w t="$." l=";">;</w>
<w t="KON" l="aber">aber</w>
<w t="ADV" l="auch">auch</w>
<w t="$." l=":">:</w>
<w t="PPER" l="er">er</w>
<w t="VVFIN" l="stellen">stelle</w>
<w t="ART" l="eine">eine</w>
<w t="ADJA" l="einfach">einfachere</w>
<w t="NN" l="Sache">Sache</w>
<w t="PTKVZ" l="dar">dar</w>
<w t="$." l=".">.</w>
</s>
```

Koreferenzaufgelöste Datei

```
<Coref> Man koennte also sagen, /
<coref id="0" key="Augustinus">Augustinus</coref> /
stelle das Lernen der Sprache zu einfach dar; /
aber auch: <coref id="0" key="Augustinus">er</coref> /
stelle eine einfachere Sache dar.</Coref>
```

Zusammengeführte Datei

```
<?xml version='1.0' encoding='ASCII'?>
<s n="Ts-213,26r[3]_4.1" ana="facs:Ts-213,26r abnr:277 satznr:741">
<w t="PIS" l="man">Man</w>
<w t="VMFIN" l="koennen">koennte</w>
<w t="ADV" l="also">also</w>
<w t="VVINF" l="sagen">sagen</w>
<w t="$," l=",">,</w>
<persName key="Augustinus, Aurelius">
<w t="NE" l="Augustinus"><coref key="Augustinus" /
keyPersName="Augustinus" isMult="0">Augustinus</coref></w>
</persName>
<choice type="s">
<seg n="s_alt1">
<w t="VVFIN" l="stellen">stelle</w>
<w t="ART" l="die">das</w>
<w t="NN" l="Lernen">Lernen</w>
<w t="ART" l="die">der</w>
<w t="NN" l="Sprache">Sprache</w>
<w t="PTKA" l="zu">zu</w>
<w t="ADJD" l="einfach">einfach</w>
<w t="PTKVZ" l="dar">dar</w>
</seg>
</choice>
<w t="$." l=";">;</w>
<w t="KON" l="aber">aber</w>
```

```
<w t="ADV" l="auch">auch</w>
<w t="$." l=":">:</w>
<w t="PPER" l="er"><coref key="Augustinus" /
keyPersName="Augustinus" isMult="0">er</coref></w>
<w t="VVFIN" l="stellen">stelle</w>
<w t="ART" l="eine">eine</w>
<w t="ADJA" l="einfach">einfachere</w>
<w t="NN" l="Sache">Sache</w>
<w t="PTKVZ" l="dar">dar</w>
<w t="$." l=".">.</w>
</s>
```

Kurzfassung

Im Satz Ts-213,26r[3]_4.1 wurde die Koreferenz `Augustinus - er` ebenso erfolgreich umgesetzt.

```
/*vor der Erweiterung */
<w t="PPER" l="er">er</w>

/*nach der Erweiterung */
<w t="PPER" l="er"><coref key="Augustinus" /
keyPersName="Augustinus" isMult="0">er</coref></w>
```

5.5 Pseudosatz mit zwei Eigennamen

Hier wird ein Satz mit 2 Personennamen und mehreren Pronomen überprüft.

Wortartgetaggte Datei

```
<s n="Pseudosatz">
<w t="KOUS" l="wenn">Wenn</w>
<persName key="Frege, Gottlob">
<w t="NE" l="Frege">Frege</w>
</persName>
<w t="APPR" l="über">über</w>
<w t="ART" l="die">die</w>
<w t="ADJA" l="formal">formale</w>
<w t="NN" l="Auffassung">Auffassung</w>
<w t="APPR" l="von">von</w>
<persName key="Russell, Betrand">
<w t="NE" l="Russell">Russell</w>
</persName>
<w t="VVFIN" l="sprechen">spricht</w>
<w t="$," l=",">,</w>
<w t="ADV" l="so">so</w>
<w t="VVFIN" l="sagen">sagt</w>
<w t="PPER" l="er">er</w>
<w t="APPR" l="über">über</w>
<w t="PPER" l="er">ihn</w>
<w t="KOUS" l="dass">dass</w>
<w t="PPER" l="er">er</w>
<w t="POSS" l="sein">sein</w>
<w t="ADJA" l="gut">guter</w>
```

```
<w t="NN" l="Freund">Freund</w>
<w t="VAFIN" l="sein">ist</w>
<w t="$." l=".">.</w>
</s>
```

Koreferenzaufgelöste Datei

```
<Coref> /
Wenn <coref id="0" key="Frege">Frege</coref> über die formale Auffassung/
von <coref id="1" key="Russell">Russell</coref> spricht,/
so sagt <coref id="0" key="Frege">er</coref> über /
<coref id="1" key="Russell">ihn</coref> /
dass <coref id="0" key="Frege">er</coref> /
<coref id="0" key="Frege">sein</coref> /
guter Freund ist.</Coref>
```

Bemerkung	id	Antezedent	Anaphora
Pseudosatz	0	key = Frege	Frege
	1	key = Russell	Russell
	0	key = Frege	er
	1	key = Russell	ihn
	0	**key = Frege**	**er**
	0	key = Frege	sein

Tabelle 5.1: Koreferenz in Pseudosatz

Die Tabelle 5.1 veranschaulicht die Grenzen der Koreferenzauflösung. Das zweite Vorkommen von `er` bezieht sich auf `Russell`. Das Tool CorZu aber markiert `Frege` als den Antezedent der Personalpronomen. Es liegt an der Gewichtung der besten Antezedentkandidaten. Laut dem Algorithmus wird im Zweifelsfall die erste, im Text erkannte nominale Phrase, als Antezedent mit dem `key` annotiert. Nur durch die manuelle Wortvektorenanpassung kann das Ergebnis verbessert werden.

Ausblick

Diese Arbeit ist die Weiterführung der Edition des Wittgenstein Nachlasses. Es ist möglich, die vorgestellten Themen weiter zu entwickeln. In diesem Kapitel wird ein Ausblick über weitere Schritte präsentiert, um die Edition des Nachlasses weiter zu verbessern.

Da es Fehler beim Wortartentagging bzw. bei syntaktischem Parsern gab, könnten andere verfügbare, für die deutsche Sprache entwickelte Tagger und Parser verwendet werden. Anschließend sollten die Ergebnisse verglichen werden.

Die verwendeten Tagger und Parser sind weiter zu entwickeln. Die genaueren morphologischen Merkmale könnten hilfreich sein, die richtigen Koreferenzkandidaten zu unterscheiden. Momentan fehlen die Möglichkeiten, verschmelzte Artikel mit Präposition, richtig zu klassifizieren. Wenn ein Verb an erster Stelle eines Satzes vorkommt, sollte dies auch als Verb erkannt werden. Es könnte kaskadierte Fehler bei der Koreferenzauflösung verhindern.

Zusätzlich dazu sollten die Editierungsfehler systematisch analysiert und behoben werden. Die Zusammenführung der Basisdatei mit der Koreferenzdatei bedarf einer Anpassung im Einzelfall. Multiple Strings, mit zwei Wörtern in der Koreferenzdatei, wurden im Rahmen dieser Arbeit implementiert. Multiple Strings mit 3 und mehr Wörtern wurden jedoch nicht berücksichtigt.

Zusammenfassung

Das Ziel, die Personennamenkoreferenzauflösung, wurde im Rahmen dieser Arbeit erfolgreich umgesetzt. Eine umfangreichere Bearbeitung aller Pronomen des Wittgenstein Nachlasses war wegen des Zeitmangels nicht umsetzbar. Dies bedarf weiterer Entwicklungen. Die Herausforderung besteht in der Natur deutscher Pronomen: sie können sich sowohl auf animierte, als auch auf unanimierte Nomen beziehen.

Zuerst war es nötig, das Koreferenzauflösungstool CorZu zu implementieren. Die Ergebnisse wurden analysiert, und eine manuelle Verbesserung der fehlerhaften Wortvektoren vorgeschlagen. Zunächst fiel der Fokus auf die Zusammenführung der ursprünglichen Basisdatei mit einer Bemerkung Wittgensteins und der koreferenzaufgelösten XML-Datei. Es wurde schnell ersichtlich, dass die multiplen Strings in der koreferenzaufgelösten XML-Datei, eine Herausforderung darstellte. Anschließend wurde das Skript von Frau Azada Rustamova optimiert. Die Evaluierung wurde abschließend auf Personennamen beschränkt. Es wurde eine Liste, sowohl mit den Personennamen, als auch mit den Pronomen, die sich auf Ludwig Wittgenstein beziehen, erstellt. Um die Erweiterung der Personensuche in der FinderApp WITTFind zu ermöglichen, wurde das Attribut `keyPersName` den relevanten Elementen zugewiesen. Damit können in Zukunft andere relevante Stellen im Nachlass gefunden werden.

Literaturverzeichnis

Faridis Alberteris Azar. Optimierung der linguistischen Suche beim XML-annotierten Nachlass von Ludwig Wittgenstein. 2017.

Max Hadersbeck, Alois Pichler, Florian Fink, Patrick Seebauer, and Olga Strutynska. *New (re) search possibilities for Wittgenstein's Nachlass*. 2012.

Max Hadersbeck, Alois Pichler, Florian Fink, and Oyvind Liland Gjesdal. Wittgenstein's Nachlass: WiTTFind and Wittgenstein advanced search tools (WAST). In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 91–96. ACM, 2014.

Stefan Langer, Petra Maier, and Jürgen Oesterle. *CISLEX-An Electronic Dictionary for German: Its Structure and a Lexicographic Application*. CIS, 1996.

Sebastian Lechner. Informationsverarbeitung, 2016. URL `https://gitlab.cis.uni-muenchen.de/dostoj-kurs/dostoj-data/tree/coref/coref`.

Sebastien Paumier, Sebastian Nagel Marschner, and Johannes Stiehler. Unitex 3.1. 2006.

Alois Pichler, H Krüger, D Smith, T Bruvik, A Lindebjerg, and V Olstad. Wittgenstein Source Bergen Facsimile Edition (BTE). *Wittgenstein Source. Bergen: WAB, Wittgenstein Source*, 2009.

Rico Sennrich and Barry Haddow. A joint dependency model of morphological and syntactic structure for statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2087, 2015.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. A new hybrid dependency parser for German. *Proceedings of the German Society for Computational Linguistics and Language Technology*, 115, 2009.

Rico Sennrich, Martin Volk, and Gerold Schneider. Exploiting synergies between open resources for German dependency parsing, pos-tagging, and morphological analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 601–609, 2013.

V. Symonenko. Disambiguierung von Partikelverb – Konstruktionen und Verbpräpositional – Konstruktionen im Nachlass von Ludwig Wittgenstein. 2015.

Don Tuggener. *Incremental coreference resolution for German*. PhD thesis, Universität Zürich, 2016.

L. Volos. Disambiguierung von Partikelverb – Konstruktionen und Verbpräpositional – Konstruktionen im Big Typescript von Ludwig Wittgenstein. 2013.

Ludwig Wittgenstein. *Tractatus logico-philosophicus*. Routledge, 2013.

Ludwig Wittgenstein, GH von Wright, et al. Wittgenstein's Nachlass the Bergen Electronic Edition. 1998.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang 1

Python Sources

```
# Author: Azada Rustamova
# Anpassung: Oksana Budurova
#
#
'''import modules'''
import sys
from lxml import etree
import re
import os
'''extract xml, coref and merged files paths'''
def parse_command():
    if len(sys.argv) != 4:
        print("Wrong number of arguments. /
        Example: python3 merge_xml_coref.py /
        <xml_file_path> <coref_file_path> <output_file_path>")
        sys.exit(1)
    xml_path = os.path.abspath(sys.argv[1])
    coref_path = os.path.abspath(sys.argv[2])
    merged_file = os.path.abspath(sys.argv[3])
    if not os.path.isfile(xml_path):
        print("Input BASIS file does not exist. /
        Please supply a valid xml file.")
        sys.exit(1)
    if not os.path.isfile(coref_path):
        print("Input COREFERENCE file does not exist. /
        Please supply a valid xml file.")
        sys.exit(1)
    return xml_path, coref_path, merged_file
'''extract information from coreference or POS_tagged node /
and return text, attribute, tag and
splitted string from node'''
def extract_node(descendants):
    word_list = []
    attribute = {}
    splt_attr = []
    for child in descendants:
        if child.text == None:
            continue
        if child.tag == "coref":
            tag = child.tag
            text = child.text
            splt_node = child.text.split()
            word_list.append(splt_node)
            ''' take a key value dictionary for attribute'''
            attrib_str = child.attrib
```

```
            for k, v in attrib_str.items():
                splt_attr.append(k)
                splt_attr.append(v)
            '''iterate over the list and add /
            the even number as key and non even as value.'''
            for i in range(len(splt_attr)):
                if i % 2 == 0:
                    name = splt_attr[i]
                    value = splt_attr[i + 1]
                    attribute[name] = value
    return text, tag, attribute, splt_node, word_list
'''function to create child'''
def create_new_child(curr_text, cur_tag, attrib_dict, idx):
    new_child = etree.Element(cur_tag)
    new_child.text = curr_text
    pronomen = ["ich","mir","mich"]
    persnamen = ['David','Frege', 'Russell', 'Augustinus', /
    'Dedekind', 'Leutnant', 'Hauptmann']
    for key in attrib_dict:
        for i in range(len(pronomen)-1):
            if attrib_dict[key] == pronomen[i]:
                attrib_dict[key]= ' Ludwig Wittgenstein'
            else:
                continue
        new_child.set(key, attrib_dict[key])
        '''it marks the personal names from the list with the attribute keyPersName'''
        for i in range(len(persnamen)-1):
            if attrib_dict[key] == persnamen[i]:
                new_child.set('keyPersName', persnamen[i])
    '''isMultiple is 0 if process_single_single_coref done, /
    else: 0 1 2 so on if process_multiple_coref done'''
    new_child.set('isMult', str(idx))
    return new_child
'''multiple_string_node with n string need to be found in the other document /
curr_node the document multiple string node needs to be found in find /
multiple_string_node[0] in curr_node. /
Multiple coref string processing: split coref multiple string, /
check if the combination of
strings matches the combination of siblings in basis xml file, /
if yes: create node with coref tag for each of /
string, and add it to the single original xml file /
with the corresponding string
'''
def process_multiple_coref(multiple_string_node, tag, attrib, curr_node):
    node_found = False
    initial_node = None
    current_string_id = 0
    while not node_found and curr_node is not None:
        '''find the first node that contains text'''
        curr_node_text = curr_node.text
        #verschachtelter Fall
        if curr_node_text is None:
            for child in curr_node.getiterator():
```

```
                curr_node_text = child.text
                if curr_node_text is not None:
                    break
        if curr_node_text == multiple_string_node[current_string_id]:
            if (current_string_id == 0):
                initial_node = curr_node
            elif (current_string_id == len(multiple_string_node) - 1):
                node_found = True
            current_string_id = current_string_id + 1
        else:
            current_string_id = 0
            initial_node = None
        curr_node = curr_node.getnext()
    '''we did not find the list of multiple_string_nodes in the document'''
    if not node_found:
        return
    curr_node = initial_node
    for i, element_text in enumerate(multiple_string_node):
        child = create_new_child(element_text, tag, attrib,i)
        curr_node.text = ""
        curr_node.append(child)
        curr_node = curr_node.getnext()
'''edit a node with single strings in both files '''
def process_single_single_coref(coref_word_lists, split_orig_node, /
orig_word, curr_orig_node, coref_tag, coref_attr):
    for coref_word_list in coref_word_lists:
        if (len(split_orig_node)) == 1 and /
        (len(coref_word_list)) == 1:
            if orig_word in coref_word_list:
                edt_single_child = /
                create_new_child(curr_orig_node.text, coref_tag, coref_attr, 0)
                curr_orig_node.text = ''
                curr_orig_node.append(edt_single_child)
                break
'''find modified node in original xml file and update it'''
def find_modified_node(orig_file, coref_file, merged_file):
    '''parse files'''
    orig_tree = etree.parse(orig_file)
    coref_tree = etree.parse(coref_file)
    '''delete all ids from coreference chain'''
    for el in coref_tree.xpath('//*[@id]'):
        el.attrib.pop('id')
    '''get all the nodes'''
    orig_descendants = orig_tree.getiterator('*')
    coref_descendants = coref_tree.getiterator('*')
    corefList = coref_tree.findall('coref')
    print("Anzahl der Koreferenzen")
    print(len(corefList))
    coref_text, coref_tag, coref_attr, splt_coref_node, /
    coref_word_lists = extract_node(coref_descendants)
    for curr_orig_node in orig_descendants:
        '''eliminate nodes without text'''
        if curr_orig_node.text == None:
```

```
            continue
        if curr_orig_node == None:
            continue
        '''split the POS_tagged node string in a list /
        for multiple string processing'''
        split_orig_node = curr_orig_node.text.split()
        '''add edited single coreference node to POS_tagged node'''
        for orig_word in split_orig_node:
            for i in corefList:
                if i.text ==orig_word:
                    process_single_single_coref(coref_word_lists,/
                    split_orig_node, orig_word, curr_orig_node, /
                    coref_tag, i.attrib)
                    break
        for coref_node in coref_word_lists:
            '''case 3: if multiple string node in unitext, /
            single string in unitext'''
            if len(coref_node) > 1:
                for i in corefList:
                    if i.text.split() == coref_node[:2]:
                        process_multiple_coref(coref_node, coref_tag, /
                        i.attrib, curr_orig_node)
    '''write new updated file'''
    orig_tree.write(merged_file, xml_declaration=True, pretty_print=True)
    print('The original POS_tagged Wittgenstein file /
    has been updated and stored in a file '+str(merged_file))


if __name__ == "__main__":
    xml_path, coref_path, merged_file  = parse_command()
    find_modified_node(xml_path, coref_path, merged_file)
```

Anhang 2

Bericht von Azada Rustamova (Squirrel Projekt XML-update)

XML Merge in the scope of Historical Linguistic Analysis

Azada Rustamova
rus.azada@gmail.com

Centrum für Informations - und Sprachverarbeitung,
Ludwig-Maximilians-Universität
Oettingenstraße 67, 1. OG, Flügel C
80538 München, Germany

1 Introduction

Latest developments in the technological area have opened new ways of performing scientific research in many social areas. The project Squirrel is a collaborative work between CIS(Centrum für Information - und Sprachverarbeitung) and the historical department of the Ludwig-Maximilians-University, which initiates integration of natural language processing (NLP) tools into the techniques of historical research.

This report provides a documentation of the implementation of XML-merge between two tools: Squirrel, a tool for historical document analysis, and Unitext, a software provided for linguistic analysis. More specifically, described program presents a method to eliminate the loss of information and keeping uniformity of data in XML documents during the transition between tools used in digital humanities.

2 Implementation

A logical structure of XML offers a hierarchical structure of information encoded in nodes. As a result, in contrast to other layout-based document formats like HTML, XML syntax allows recording semantics and structure in the documents.

However, the logical and hierarchical structure of XML format poses challenges in performing changes in the tree structure of documents. Overlapping is one of the most challenging questions that arise during XML processing. This project offers a solution to updating original XML document in the scope of the hierarchical structure of the document, and overlapping that poses additional challenges to the XML merge.

The implementation of the program provided methods to preserve XML node structure in the original Squirrel XML document and add additional node information from the Unitext XML document. Unitext outputs a .txt document, which first should be transferred into information by adding an xml opening and closing tags the beginning and at the end of the document. After transforming .txt document into .xml format, the merge between Squirrel .xml and UNITEXT

.xml document can be performed. The merge involves iterating over strings in both documents, comparing them and updating original squirrel node tree in the case of a modification in the Unitext .xml file.

As we already have mentioned, overlapping of node trees in two documents poses additional difficulties during the merge. Firstly, we addressed the issue of multiple string node in the original file where we split text of every node and then transfer each string into a separate child with node information. Every newly created node would possess attribute information of the parenting node and will, hence, update node information in the Squirrel document. Multiple strings on Unitext file make the transition more difficult and require a recursive search and update in the original file. A python lxml package was used as a framework for this implementation.

3 Future Work and Conclusion

This project is an initial step in integrating linguistic tools into historical research methodology. Despite such challenges as overlapping, we have provided solutions for XML merge that is an integral part of transferring linguistic information into the historical document. Two cases of overlapping node structure were analyzed and eliminated. An example with multiple strings in both documents having different start and end tags was not in the scope of this project. Moreover, the nested nodes that have textual information in the leaf nodes are requiresively searched and extracted, however, their update requires more elaborate work in the future.

Inhalt der beigelegten CD

- Abschlussarbeit

  beinhaltet eine elektronische Version der Bachelorarbeit

  - Latex Dateien (.tex-Datei, .sty-Datei und .bib-Datei)
  - PDF Datei

- Code

  beinhaltet die Software und die erstellten Skripte

  - Readme.md: gibt einen Überblick über den Inhalt der CD
  - ext: beinhaltet externe Software (CorZu, Bericht von Azada Rustamova)
  - lib: beinhaltet sowohl ein Datenanpassungs- als auch ein Mergingmodul (deleteSP.sh, xmlToText.py und merge_POS_coref.py)
    - eval: beinhaltet Ressourcen für die Evaluierung (.txt und .xml Dateien)
    - dev: beinhaltet Ressourcen für die Entwicklung (.txt und .xml Dateien)
  - stat: beinhaltet Skripte für die Erstellung der Frequenz der Koreferenzen, .csv-Dateien und Bar Charts im .jpeg-Format
  - examples: beinhaltet Beispiele für die Präsentation dieser Arbeit