Kombination mehrerer lexikalisch-semantischer Ressourcen durch multiple Alignments von Wortbedeutungen

Combining multiple lexical-semantic resources using multiple word sense alignments

Master-Thesis von Christian Kirschner

4. Dezember 2012

TECHNISCHE
UNIVERSITÄT
DARMSTADT

UBIQUITOUS
KNOWLEDGE
PROCESSING

Kombination mehrerer lexikalisch-semantischer Ressourcen durch multiple Alignments von Wortbedeutungen
Combining multiple lexical-semantic resources using multiple word sense alignments

vorgelegte Master-Thesis von Christian Kirschner

Supervisor: Prof. Dr. Iryna Gurevych
Coordinator: Christian M. Meyer

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 4. Dezember 2012

_____

(Christian Kirschner)

Zusammenfassung

Viele Anwendungen aus der natürlichen Sprachverarbeitung wie automatische Textzusammenfassung oder maschinelle Übersetzung bauen auf lexikalisch-semantischen Ressourcen auf. In dieser Masterarbeit beschäftigen wir uns mit der Kombination von mehr als zwei lexikalisch-semantischen Ressourcen wie WordNet, Wikipedia, Wiktionary und OmegaWiki indem wir übereinstimmende Wortbedeutungen aus diesen Ressourcen einander zuweisen (multiples Alignment). Ziel dieser Arbeit ist es, die Grundlagen multipler Alignments zu erforschen und zu prüfen, ob es möglich ist, mit einem multiplen Alignment eine höhere Qualität zu erreichen als mit paarweisen Alignments. Bei letzteren werden die Wortbedeutungen aus nur genau zwei Ressourcen einander zugewiesen. Eine höhere Qualität erhoffen wir uns durch das Ausnutzen der globalen Struktur, welche durch die Verwendung von mehr als zwei Ressourcen entsteht.

Wir stellen in dieser Arbeit zwei Ansätze zur Berechnung multipler Alignments vor: Der konstruktive Ansatz berechnet paarweise Ähnlichkeitswerte zwischen den Wortbedeutungen aus den verschiedenen Ressourcen und identifiziert mit Clustering-Algorithmen Gruppen von ähnlichen Wortbedeutungen. Der korrektive Ansatz hingegen baut auf den bereits vorhandenen paarweisen Alignments auf und versucht diese durch eine Fehlerkorrektur zu verbessern. Die Evaluation zeigt, dass beide Ansätze das Potenzial besitzen die Qualität einfacher paarweiser Alignments zu übertreffen.

Abstract

There are a lot of applications in Natural Language Processing such as automatic text summarization or machine translation which build on lexical-semantic resources. In this master thesis we concentrate on combining more than two lexical-semantic resources as WordNet, Wikipedia, Wiktionary and OmegaWiki by aligning accordant word senses from these resources (multiple alignment). The purpose of this elaboration is to explore the basics of multiple alignments and to investigate if it is possible to achieve a higher quality in a multiple alignment in comparison with a pairwise alignment which aligns word senses from exactly two resources. We hope to get a higher quality by exploiting of the global structure which arises by using more than two resources.

In this elaboration we present two approaches to calculate multiple alignments: The constructive approach calculates pairwise similarity values between the word senses from different resources and identifies groups of similar word senses with the help of clustering algorithms. The corrective approach on the other hand bases upon existing pairwise alignments and tries to improve those alignments with the help of an error correction. The evaluation indicates that both approaches have the potential to outperform the quality of simple pairwise alignments.

Inhaltsverzeichnis

1 Einleitung

In den vergangenen Jahren hat sich das Internet zu einer gewaltigen Ansammlung von größtenteils unstrukturierten Daten entwickelt. Das Forschungsgebiet „Natural Language Processing“ (NLP) beschäftigt sich unter anderem mit der computergestützten Erkennung der Bedeutung von Texten. Zu den wichtigsten Forschungsfeldern zählen Word Sense Disambiguation (WSD), automatische Textzusammenfassung oder maschinelle Übersetzung, wobei die beiden zuletzt genannten Anwendungen auf Word Sense Disambiguation aufbauen.

Das Ziel dieser Arbeit ist im weitesten Sinne eine Verbesserung dieser Anwendungen zu erreichen. Dazu beschäftigen wir uns mit dem Alignment von Wortbedeutungen. Darunter ist zu verstehen, dass wir übereinstimmende Wortbedeutungen aus unterschiedlichen Quellen und mit verschiedenen Beschreibungen einander zuweisen. Als Datenquellen für Wortbedeutungen dienen sogenannte lexikalisch-semantische Ressourcen, die im Wesentlichen aus einer Auflistung von Wörtern und deren möglichen Bedeutungen sowie Informationen bezüglich der Beziehungen zwischen den einzelnen Bedeutungen (semantische Beziehungen) bestehen. Diese Ressourcen werden in den NLP-Anwendungen genutzt um beispielsweise die Bedeutung eines Wortes innerhalb eines Satzes zu erkennen.

Ein solches Alignment von Wortbedeutungen aus verschiedenen Ressourcen ermöglicht die Kombination der entsprechenden Ressourcen, sodass die darauf aufbauenden NLP-Anwendungen die gebündelten Informationen aus mehreren Ressourcen gleichzeitig nutzen können. So können wir die Heterogenität der Ressourcen nutzen um eine höhere Abdeckung an Bedeutungen zu erhalten (Bedeutungen, die nur in einer der Ressourcen vorkommen). Gleichzeitig können wir durch das Alignment übereinstimmende Wortbedeutungen durch das Zusammenführen mit den zugehörigen Informationen aus den verschiedenen Ressourcen anreichern und neue semantische Relationen finden. Verschiedenste Arbeiten aus dem Bereich semantisches Parsen [Shi and Mihalcea, 2005], Word Sense Disambiguation [Ponzetto and Navigli, 2010] oder multimodale Datenbanken [de Melo and Weikum, 2010] haben bestätigt, dass die Arbeit auf einer erweiterten Ressource zu besseren Ergebnissen führen kann.

Das Alignment von Wortbedeutungen aus genau zwei lexikalisch-semantischen Ressourcen bezeichnen wir als paarweises Alignment. Diesbezüglich wurden bereits zahlreiche Ansätze veröffentlicht (Niemann and Gurevych [2011], Meyer and Gurevych [2011], Shi and Mihalcea [2005], Johansson and Nugues [2007], etc.). In dieser Arbeit gehen wir über paarweise Alignments hinaus, indem wir uns mit der Alignierung von Wortbedeutungen aus mehr als zwei Ressourcen auseinandersetzen, was wir folglich als multiples Alignment bezeichnen. Unseres Wissens gibt es bislang keine Arbeiten, die sich mit dieser Problemstellung befasst haben. Neben einer noch reichhaltigeren Ressource erhoffen wir uns insbesondere eine höhere Qualität der automatisch berechneten Alignments. Während bei einem paarweisen Alignment grundsätzlich nur die Ähnlichkeit zweier Wortbedeutungen über deren Alignment entscheidet, entsteht in einem multiplen Alignment durch die Betrachtung von mehr als zwei Ressourcen eine globale Struktur, die es ermöglicht die in paarweisen Alignments gemachten Fehler zu vermeiden.

In Kapitel 2 werden wir einige wichtige Begriffe zum Alignment von Wortbedeutungen erläutern und die in dieser Arbeit genutzten Ressourcen vorstellen. Kapitel 3 gibt einen Überblick über verwandte Arbeiten zu paarweisen und multiplen Alignments. Anschließend führen wir in Kapitel 4 eine quantitative und qualitative Datenanalyse der Daten aus den in Kapitel zwei vorgestellten Ressourcen durch. In Kapitel 5 stellen wir Ansätze für die Berechnung multipler Alignments vor, die wir in Kapitel 6 evaluieren. Wir schließen mit einer Zusammenfassung in Kapitel 7.

2 Grundlagen

In diesem Kapitel werden wir zunächst die Grundlagen paarweiser und multipler Alignments sowie deren Zusammenhang behandeln, die wichtigsten Unterschiede erläutern und einige zentrale Begriffe definieren. Anschließend gehen wir auf die Ressource UBY ein, die Informationen aus unterschiedlichen lexikalisch-semantischen Ressourcen wie WordNet, Wikipedia und Wiktionary enthält und die Datengrundlage für die im Folgenden entwickelten Verfahren darstellt.

2.1 Alignments von Wortbedeutungen

Wie in der Einleitung geschildert, geht es in dieser Arbeit um die Kombination mehrerer lexikalisch-semantischer Ressourcen durch multiple Alignments von Wortbedeutungen. Unter einem Alignment verstehen wir die Zuweisung von übereinstimmenden bzw. synonymen Wortbedeutungen, in der Regel aus unterschiedlichen Ressourcen. Während bei einem paarweisen Alignment Wortbedeutungen aus genau zwei Ressourcen einander zugewiesen werden, arbeiten wir bei einem multiplen Alignment mit Wortbedeutungen aus mehr als zwei Ressourcen.

Der Begriff „Wortbedeutung“ entspricht dem häufig genutzten Begriff „Sense“ (oder „Word Sense“, z.B. Jurafsky and Martin [2000]). Ein Sense ist durch ein Wort und dessen zugehörige Bedeutung identifiziert. Ein Wort kann verschiedene Senses haben (z.B. Bank im Sinne von Geldinstitut und Bank im Sinne von Sitzbank). Zwei oder mehr verschiedene Wörter stellen immer auch verschiedene Senses dar, auch wenn sie die gleiche Bedeutung haben (z.B. Auto und Automobil). Im Kontext der lexikalisch-semantischen Ressource WordNet [Fellbaum, 1998] wurde außerdem der Begriff des „Synsets“ erschaffen: Ein Synset („set of synonyms“) fasst alle synonymen Wörter einer Ressource (Wörter, die eine bestimmte Bedeutung teilen) zusammen. Unterschiedliche Wörter, die jedoch die gleiche Bedeutung beschreiben (z.B. Auto und Automobil), bilden folglich ein Synset. Jedes zu einem Synset gehörende Synoymwort stellt einen eigenen Sense dar. Man könnte daher auch sagen, dass eine ressourceninterne Zuweisung von synonymen Senses (also quasi ein ressourceninternes Alignment) diese in Synsets überführt, weil dabei alle Wörter mit der gleichen Bedeutung in Synsets zusammengefasst werden.

Senses sehen wir dann als synonym an, wenn sie die gleiche Bedeutung beschreiben: Die Einträge „Apple: fruit with red or yellow or green skin and sweet to tart crisp whitish flesh“ aus WordNet und „Apple: The apple is the pomaceous fruit of the apple tree (...)“ aus Wikipedia sind dementsprechend synonym und einander zuzuweisen. Ebenso synonym ist der Wiktionary Eintrag „Apple: a common, round fruit produced by the tree Malus domestica (...)“ (siehe Abbildung 2.1).

Da wir in einem multiplen Alignment mit mehr als zwei Ressourcen arbeiten, müssen wir auch mehr als zwei Senses einander zuweisen können. Ein multiples Alignment von drei Senses lässt sich daher durch drei paarweise Alignments ausdrücken. Folglich werden in einem multiplen Alignment grundsätzlich sämtliche Senses paarweise aligniert, woraus folgt, dass sämtliche Senses in einem multiplen Alignment synonym sein müssen. Um diese Eigenschaft sicherzustellen definieren wir Synonymie als reflexiv, symmetrisch und transitiv. Werden in einem multiplen Alignment die Senses A und B, sowie die Senses B und C einander zugewiesen, so müssen entsprechend der Transitivitätseigenschaft dadurch immer auch die Senses A und C einander zugewiesen sein (siehe Abbildung 2.1).

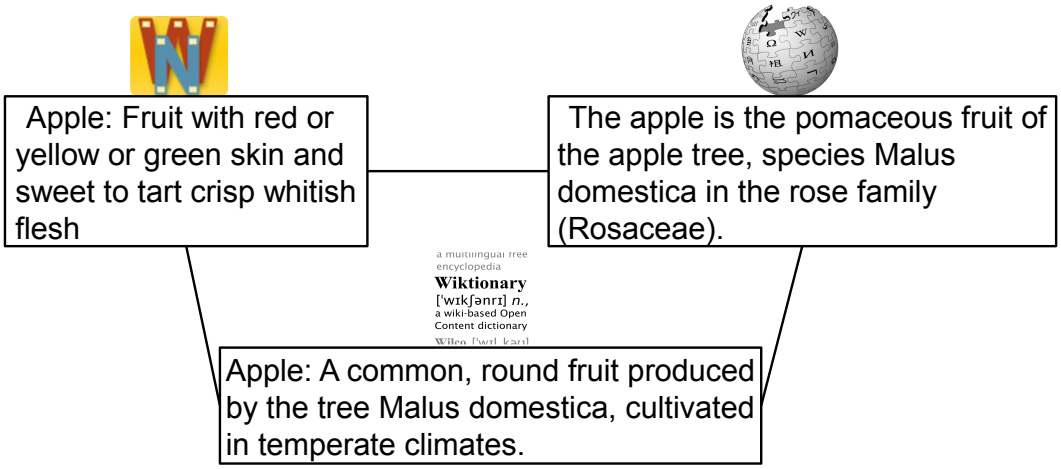


Abbildung 2.1: Eine multiples Alignment von 3 Senses (aus Wikipedia, WordNet und Wiktionary), ausgedrückt durch 3 paarweise Alignments

Graphen sind eine geeignete Visualisierung für multiple Alignments, die wir im Folgenden immer wieder nutzen werden. Wir zitieren an dieser Stelle daher einige für diese Arbeit relevante Definitionen aus Tittmann [2003]: „Ein ungerichteter Graph G = (V, E) besteht aus einer Knotenmenge V und einer Kantenmenge E, wobei jeder Kante e ∈ E von G zwei (...) Knoten aus V zugeordnet sind“ (S. 12). „Ein isolierter Knoten ist ein Knoten vom Grade null. Ein isolierter Knoten besitzt keine Nachbarknoten“ (S. 13). „Ein Graph G = (V, E) heißt zusammenhängend, wenn zwischen je zwei Knoten u und v seiner Knotenmenge ein Weg existiert. Ein maximaler zusammenhängener Untergraph eines Graphen heißt eine Komponente von G“ (S. 15). „Ein vollständiger Graph K_n mit n Knoten besitzt zwischen je zwei seiner Knoten genau eine Kante“ (S. 20). In Bezug auf unsere Problemstellung stellen die Knoten eines Graphen die Senses aus unterschiedlichen Ressourcen dar, die Kanten zwischen den Knoten entsprechen in ungewichteter Form Zuweisungen aus paarweisen Alignments oder geben in gewichteter Form die Ähnlichkeit der sie verbindenden Senses an.

Als Hauptmotivation für multiple Alignments haben wir in der Einleitung einen zu erwartenden Qualitätsvorteil genannt, welcher sich aus der durch mehr als zwei Ressourcen entstehenden globalen Struktur ergibt. Um dies näher zu begründen ist ein Blick auf die Vorgehensweise bei der Berechnung paarweiser Alignments notwendig: Um zu entscheiden, ob zwei Senses unterschiedlicher Ressourcen in einem paarweisen Alignments einander zugewiesen werden sollten, wird für die beiden Senses ein Ähnlichkeitswert berechnet. Dieser ergibt sich in der Regel aus einem Vergleich der Beschreibungstexte zu den zu vergleichenden Senses. Teilweise werden auch noch weitere Informationen hinzugezogen, der Vergleich kann sowohl syntaktisch als auch semantisch sein. Sofern der Ähnlichkeitswert einen zuvor trainierten Schwellenwert übersteigt, findet dann eine Zuweisung der beiden Senses statt.

In dem Graphen aus Abbildung 2.2 beschreiben die beiden linken Knoten die Bedeutung des „Föderalismus“, während die drei rechten Knoten für die Bedeutung der „Föderation“ stehen. Bei einem Schwellenwert von 0,5 ergäben in einem paarweisen Alignment die in der Abbildung rot markierten Kanten jeweils eine Zuweisung, da der Ähnlichkeitswert hier über dem Schwellenwert liegt. Dadurch entsteht jedoch eine fehlerhafte Zuweisung zwischen den Bedeutungen „Föderalismus“ und „Föderation“. Betrachten wir Alignments als transitiv, so bewirkt dieser eine Fehler, dass alle fünf Senses im Graphen als synonym angesehen werden, da der Graph zusammenhängend ist.

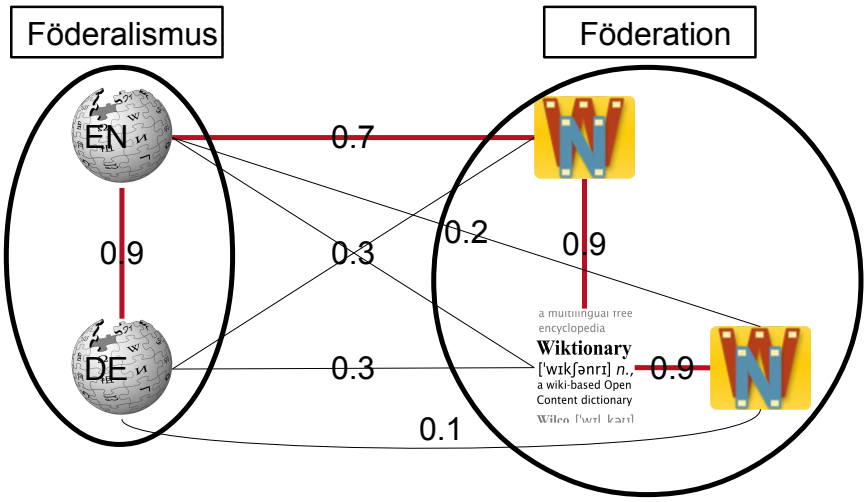


Abbildung 2.2: Visualisierung von Senses und deren Beziehungen als Graph

Während wir bei einem paarweisen Alignment für die Zuweisung folglich immer nur eine einzelne Kante (lokal) berücksichtigen, können wir bei einem multiplen Alignment die Gesamtstruktur (global) einbeziehen. Dadurch, dass es (neben der einen Kante mit hohem Gewicht) viele Kanten mit sehr geringem Gewicht zwischen den beiden Bedeutungen (Föderalismus und Föderation) gibt, können wir erkennen, dass hier zwei verschiedene Bedeutungen vorliegen und die fehlerhafte Zuweisung somit vermeiden. Wir benötigen zur Berechnung multipler Alignments somit Algorithmen, die einen zusammenhängenden Graphen entsprechend der Kantengewichte in stark zusammenhängende Cluster unterteilen. Zu diesem Zweck eignen sich insbesondere Clustering-Algorithmen, die wir in Abschnitt 5.1.4 vorstellen.

2.2 Die Ressource UBY

In diesem Abschnitt wird die Ressource UBY[1] vorgestellt [Gurevych *et al.*, 2012]. UBY ist eine an der TU Darmstadt entwickelte großangelegte lexikalische Ressource, die Informationen aus insgesamt neun Ressourcen in einem standardisierten Format enthält (siehe Abbildung 2.3). Darunter sind sowohl englisch- als auch deutschsprachige Ressourcen. Dazu werden die Informationen aus den verschiedenen Ressourcen in Form von Senses gespeichert. Die Senses der verschiedenen Ressourcen sind zum Teil über paarweise Alignments miteinander verknüpft, was insbesondere für diese Arbeit eine wichtige Datengrundlage darstellt. Abbildung 2.3 gibt einen Überblick über die in UBY enthaltenen Ressourcen, die Anzahl der enthaltenen Senses, sowie über vorhandene paarweise Alignments zwischen den Senses der Ressourcen. Wir beschränken uns auf die in Abbildung 2.3 nicht schraffierten sechs Ressourcen WordNet[2] (englisch), Wikipedia[3] (deutsch + englisch), Wiktionary[4] (englisch) und OmegaWiki[5] (deutsch + englisch) und die sie verbindenden sieben paarweisen Alignments. Die Ressourcen FrameNet[6] und VerbNet[7] betrachten wir in dieser Arbeit nicht näher, da sie sich in ihrem Aufbau recht stark von den anderen Ressourcen unterscheiden: FrameNet beispielsweise baut nicht auf einzelnen

[1] http://www.ukp.tu-darmstadt.de/data/lexical-resources/uby
[2] http://wordnet.princeton.edu/
[3] http://www.wikipedia.org/
[4] http://www.wiktionary.org/
[5] http://www.omegawiki.org/
[6] http://framenet.icsi.berkeley.edu/
[7] http://verbs.colorado.edu/ mpalmer/projects/verbnet.html

Wörtern, sondern auf sogenannten „semantischen Frames“ auf, die Ereignisse, Beziehungen und Zustände charakterisieren [Baker and Fellbaum, 2009]. Die deutsche Wiktionary schließen wir aus, da es in UBY bislang keine paarweisen Alignments zu dieser Ressource gibt.

Ziel dieser Arbeit ist es die Grundlagen für die Berechnung eines multiplen Alignments zu erforschen. Bei einem multiplen Alignment werden die übereinstimmenden Senses aus drei oder mehr Ressourcen einander zugewiesen. Bei der Entwicklung und Evaluation der in dieser Arbeit vorgestellten Verfahren zur Berechnung eines solchen multiplen Alignments (siehe Kapitel 5) dienen die in UBY enthaltenen Ressourcen als Datengrundlage.

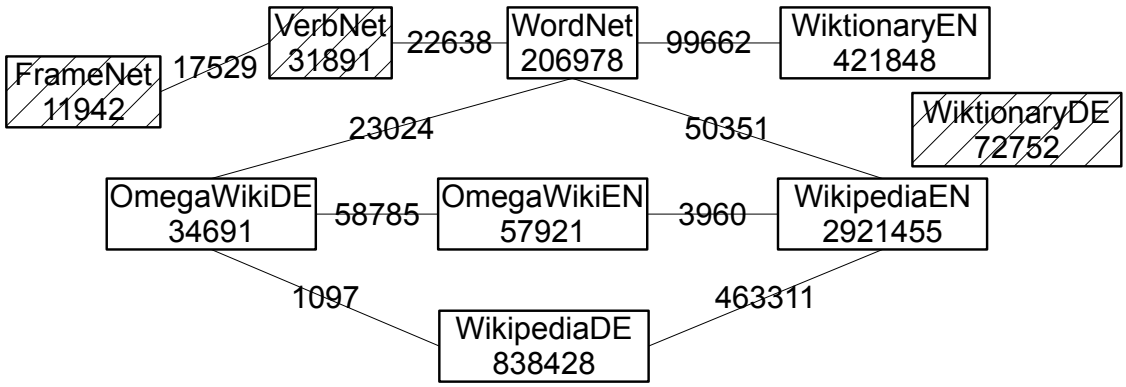


Abbildung 2.3: In UBY enthaltene Ressourcen und die Anzahl der darin enthaltenen Senses, sowie Anzahl der Zuweisungen aus paarweisen Alignments

Im Folgenden werden die verschiedenen von uns betrachteten, in UBY enthaltenen Ressourcen vorgestellt und Unterschiede besprochen.

2.2.1 WordNet

Die lexikalische Wissensbasis WordNet [Fellbaum, 1998] ist das in NLP am häufigsten genutzte englischsprachige semantische Wörterbuch. Die Bedeutungen werden durch insgesamt 117.659 Synsets (Version 3.0) repräsentiert. Ein Synset enthält in WordNet die der Bedeutung zugehörigen Synonymwörter, eine kurze Beschreibung (Gloss) und teilweise einen kleinen Beispielsatz, außerdem verschiedene semantische und lexikalische Beziehungen zu anderen Synsets (Hyponym, Hyperonym, Meronym, Antonym etc.). Es gibt Synsets zu den Wortarten Nomen, Verb, Adjektiv und Adverb. WordNet wurde von Linguisten entwickelt, was für eine hohe Qualität der Daten spricht, und ist frei verfügbar.

Zu den Stärken von WordNet zählen die Berücksichtigung der vier Wortarten Nomen, Verb, Adjektiv und Adverb sowie die semantischen Relationen, die für viele NLP Anwendungen von großer Relevanz sind. So beabsichtigen wir durch die Kombination mehrerer Ressourcen neben einer größeren Abdeckung und einem höheren Informationsgehalt auch neue semantische Relationen zu finden. Von Nachteil ist die zu einem Synset angebotene Informationsmenge, die sich in der Regel auf ein bis zwei kurze Sätze (Gloss) beschränkt. Außerdem ist die nicht optimale Abdeckung der Senses sowie die mangelnde Aktualität anzumerken. So fehlen beispielsweise aktuelle Begriffe wie „firefox“, oder „perl“ [Meyer and Gurevych, 2010]. In UBY sind neben den WordNet Synsets auch die zugehörigen Senses gespeichert. Jedes in einem Synset enthaltene Synonymwort stellt einen eigenen Sense in UBY dar.

2.2.2 Wikipedia

Wikipedia hat große Bekanntheit als frei verfügbare, gemeinschaftlich entwickelte Online Encyclopädie erlangt, deren Inhalte von hoher Qualität sind [Giles, 2005]. Senses werden durch die einzelnen Artikel repräsentiert, Synonyme lassen sich mit sogenannten Redirects aufspüren, semantische Beziehungen erhält man über den Kategoriegraphen und die In-/Outlinks eines Artikels. Da die Daten zunächst jedoch in unstrukturierter Form vorliegen, sind entsprechende Verarbeitungsschritte notwendig, um diese semantischen Informationen nutzen zu können.

Mit den mittlerweile 3.993.083 englischsprachigen und 1.430.440 deutschsprachigen Artikeln (Juli 2012) erreicht Wikipedia eine deutlich höhere Abdeckung an Senses als alle anderen hier betrachteten Ressourcen und enthält zudem eine sehr große Menge enzyclopädischen Wissens zu den einzelnen Senses. Im Unterschied zu WordNet werden auch sehr aktuelle Senses (z.B. über Filme oder Personen) abgedeckt, es werden jedoch nahezu ausschließlich Nomen beschrieben. Von großem Nutzen für viele Anwendungen ist zudem die Multilingualität von Wikipedia (z.B. Potthast *et al.* [2008]).

In UBY entspricht jeder Wikipedia Artikel einem Sense, für die Beschreibung des Senses dient der erste Absatz jedes Artikels (siehe Abbildung 2.4). Obwohl über Redirects mehrere synonyme Wörter auf den gleichen Artikel verweisen können, ist in UBY lediglich ein Sense pro Artikel enthalten: So gibt es beispielsweise den UBY Sense „Automobile“, nicht jedoch den UBY Sense „Car“, obwohl ein Redirect auf den Artikel „Automobile“ verweist. Während wir folglich mehrere UBY Senses aus WordNet mit der gleichen Bedeutung haben können, ist dies bei Wikipedia nicht der Fall.

> A **table** is a form of furniture with a flat and satisfactory horizontal upper surface used to support objects of interest, for storage, show, and/or manipulation.[1] The surface must be held stable; for reasons of simplicity, this is done by support from below by either a column, a "base" or at least three columnar "stands".

Abbildung 2.4: Erster Absatz eines Wikipedia Artikels

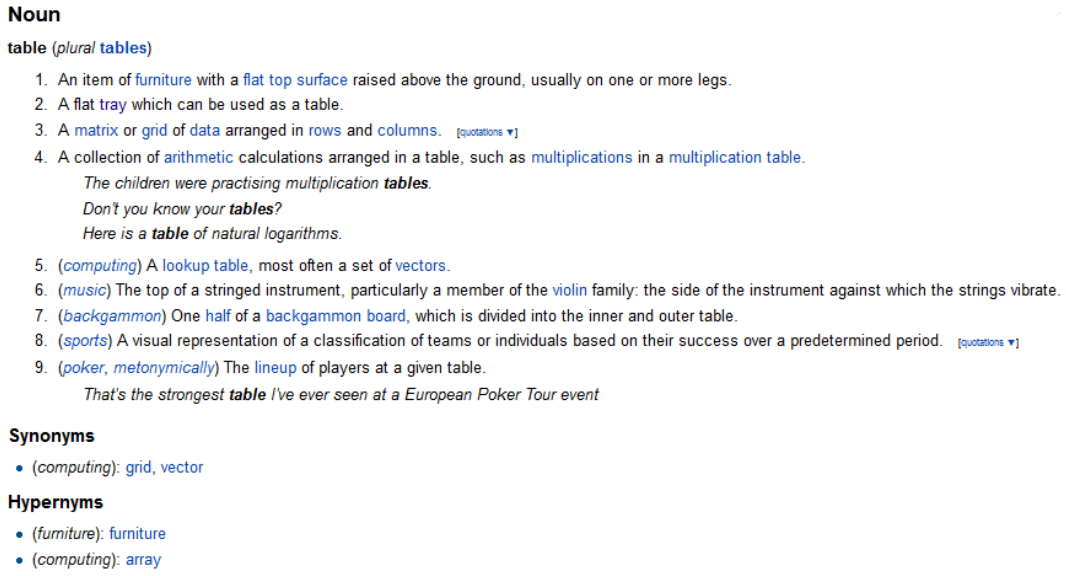
2.2.3 Wiktionary

Bei Wiktionary handelt es sich um ein frei verfügbares, gemeinschaftlich entwickeltes, mehrsprachiges Online-Wörterbuch mit semantischen Beziehungen. Die englischsprachige Version enthält mittlerweile über 3 Millionen Artikel, bei der deutschsprachigen Version sind es aktuell 200.000 Artikel (Juli 2012). Senses werden durch eine Beschreibung (Gloss), Synonyme und eventuell Beispielsätze repräsentiert. Außerdem gibt es semantische Relationen wie Hyperonyme und Hyponyme, sowie multilinguale Verknüpfungen in Form von Übersetzungen für Wörter (siehe Abbildung 2.5). Diese semantischen Relationen sind jedoch nicht für alle Senses vollständig vorhanden.

Es ist hervorzuheben, dass ein Artikel nicht der Bedeutung eines Wortes bzw. einem Sense entspricht (wie in Wikipedia). Vielmehr enthält jeder Artikel alle möglichen Bedeutungen (Senses) eines Wortes. Dies hat zur Folge, dass die gleiche Bedeutung auf unterschiedliche Weise in verschiedenen Artikeln beschrieben wird. Die Bedeutung der synonymen Wörter „actor“ und „performer“ wird beispielsweise in verschiedenen Artikeln auf verschiedene Art und Weise umschrieben: „A person who performs in a theatrical play or film“ bzw. „One who performs for, or entertains, an audience“. Anders als bei WordNet ist das Wörterbuch also in Form von Senses strukturiert. Eine Bedeutung wird daher häufig

durch mehrere Senses repräsentiert, was zunächst mal der Sense-Definition entspricht. Da die Senses, auch wenn sie die gleiche Bedeutung haben, jedoch unterschiedliche Beschreibungen besitzen, ist das Zusammenfassen von Senses zu Synsets hier im Vergleich zu beispielsweise WordNet deutlich erschwert.

Wie auch bei Wikipedia ist die Mehrsprachigkeit und die Abdeckung von (auch aktuellen) Wörtern positiv zu bewerten, wobei im Unterschied zu Wikipedia keine Personen, aktuelle Filme oder Sportereignisse enthalten sind. Als Nachteil könnte sich die angesprochene Strukturierung nach Senses (nicht nach Synsets wie bei WordNet) herausstellen. Die semantischen Relationen (Hyperonyme, Hyponyme) sind nicht immer vollständig vorhanden und beziehen sich zudem auf andere Wörter und nicht auf andere Senses (insbesondere in der englischen Wiktionary). Wenn wir also beispielsweise den Sense „actor: a person who performs in a theatrical play or film“ betrachten, dann ist als Synonymwort das Wort „performer“ angegeben. Da aber das Wort „performer“ unterschiedliche Bedeutungen und somit unterschiedliche Senses haben kann, ist unklar auf welchen Sense sich die Relation bezieht, was eine automatisierte Nutzung der semantischen Relationen durch NLP Anwendungen erschwert.



Noun

table (*plural* **tables**)

1. An item of furniture with a flat top surface raised above the ground, usually on one or more legs.
2. A flat tray which can be used as a table.
3. A matrix or grid of data arranged in rows and columns. [quotations ▾]
4. A collection of arithmetic calculations arranged in a table, such as multiplications in a multiplication table.
   *The children were practising multiplication **tables**.*
   *Don't you know your **tables**?*
   *Here is a **table** of natural logarithms.*
5. (*computing*) A lookup table, most often a set of vectors.
6. (*music*) The top of a stringed instrument, particularly a member of the violin family: the side of the instrument against which the strings vibrate.
7. (*backgammon*) One half of a backgammon board, which is divided into the inner and outer table.
8. (*sports*) A visual representation of a classification of teams or individuals based on their success over a predetermined period. [quotations ▾]
9. (*poker*, *metonymically*) The lineup of players at a given table.
   *That's the strongest **table** I've ever seen at a European Poker Tour event*

Synonyms

- (*computing*): grid, vector

Hypernyms

- (*furniture*): furniture
- (*computing*): array

Abbildung 2.5: Ausschnitt eines Wiktionary-Artikels

2.2.4 OmegaWiki

OmegaWiki ist ähnlich wie Wiktionary ein frei verfügbares, gemeinschaftlich entwickeltes mehrsprachiges Online-Wörterbuch mit semantischen Beziehungen. Im Gegensatz zu Wiktionary sind die Einträge jedoch in Form von Synsets angeordnet. Außerdem vereint OmegaWiki alle Sprachen in einem einzigen Wörterbuch. Die Synsets fassen daher Synonymwörter (und Beschreibungstexte) unterschiedlicher Sprachen zusammen, wodurch eine sehr nützliche multilinguale Ressource entsteht (siehe Abbildung 2.6). Enthalten sind außerdem semantische Relationen wie Hyperonyme und Hyponyme und Klassenzugehörigkeit (beispielsweise „Beruf“ für Schauspieler). Leider ist die Anzahl der enhaltenen Senses im Vergleich zu anderen betrachteten Ressourcen noch recht klein. Für UBY werden die Synsets wiederum nach ihren Sprachen zerteilt und mit Hilfe der Synonymwörter in Senses zerlegt. Die Relationen zwischen den Synsets der Sprachen Englisch und Deutsch fließen als multilinguales paarweises Alignment von sehr hoher Qualität in UBY ein.

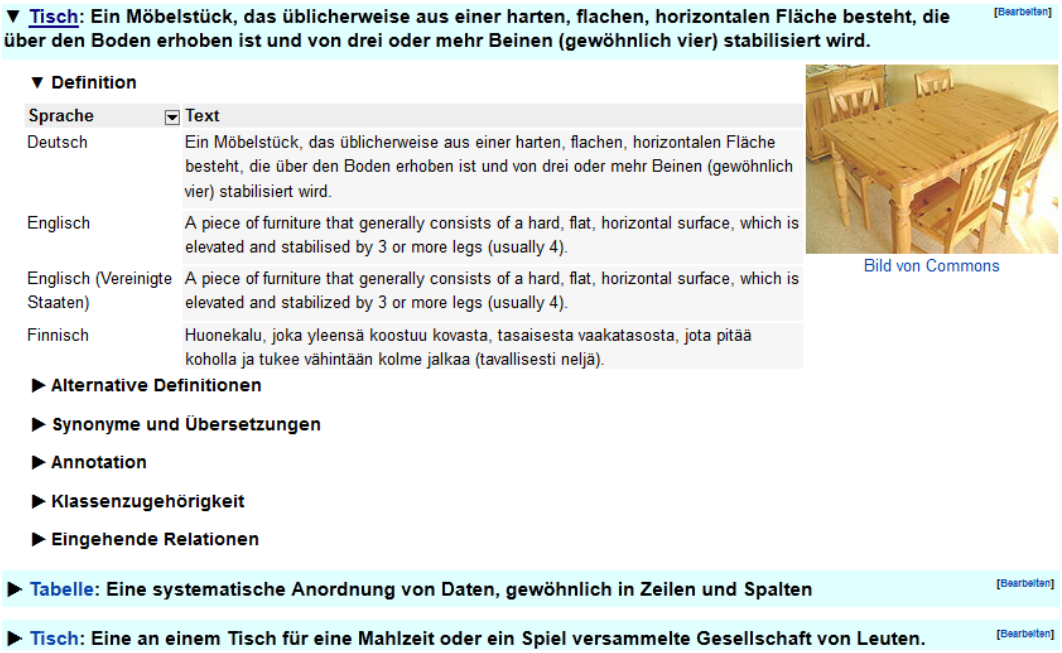


Abbildung 2.6: OmegaWiki Eintrag zu dem Begriff „Tisch“

2.2.5 Gegenüberstellung der Ressourcen

Wie bereits erwähnt unterscheiden sich die Ressourcen in den angebotenen Informationen deutlich. Diese Heterogenität kann genutzt werden um die Abdeckung an Senses zu erhöhen, Senses mit Informationen anzureichern und neue semantische Relationen zu finden. Tabelle 2.1 gibt eine Übersicht über die Stärken der verschiedenen vorgestellten Ressourcen. Wir erkennen, dass jede der vier Ressourcen in unterschiedlichen Bereichen Stärken aufweist, sodass die Ressourcen sich gut ergänzen können.

	WordNet	Wikipedia	Wiktionary	OmegaWiki
Wortarten	+	−	+	+
Anzahl Senses	o	+	o	−
Informationsmenge	−	+	−	−
Aktualität	−	+	o	o
Semantische Relationen	+	−	o	+
Multilingualität	−	o	o	+
Strukturierung (Synsets)	+	o	−	+

Tabelle 2.1: Gegenüberstellung der verschiedenen Ressourcen

3 Verwandte Arbeiten

Dieses Kapitel beleuchtet verwandte Arbeiten aus dem Bereich der Alignierung von Wortbedeutungen. Bezüglich der Berechnung paarweiser Alignments von Wortbedeutungen existiert bereits eine Vielzahl an Ansätzen. Die Berechnung multipler Alignments von Wortbedeutungen wurde hingegen noch nicht näher betrachtet. Lediglich aus anderen Bereichen wie der Bioinformatik sind multiple Alignments bekannt. Im Folgenden werden wir zunächst einige wichtige Arbeiten zu paarweisen Alignments aufgreifen und anschließend auf multiple Alignments eingehen.

3.1 Paarweise Alignments von Wortbedeutungen

Aus den zahlreichen Arbeiten zu paarweisen Alignments von Wortbedeutungen wollen wir in diesem Abschnitt einige Arbeiten herausgreifen, die sich mit der Kombination von zwei der in Kapitel 2 vorgestellten Ressourcen beschäftigen.

Suchanek *et al.* [2007] konstruieren eine Wissensbasis („YAGO“ - Yet Another Great Ontology) mit Hilfe von Informationen aus den Ressourcen WordNet und Wikipedia. Allerdings findet die Kombination dieser beiden Ressourcen hier nicht in dem Sinne statt, dass übereinstimmende Wortbedeutungen der beiden Ressourcen einander zugewiesen werden. Stattdessen enthält die erstellte Wissensbasis sämtliche WordNet Synsets der Wortart Nomen und ergänzend dazu alle Wikipedia Artikel, deren Titel nicht bereits durch ein WordNet Synset abgedeckt sind. Das Ziel dieser Vorgehensweise ist es, die in WordNet häufig fehlenden Einträge über Personen oder Orte durch die Informationen aus Wikipedia zu ergänzen. Dadurch gehen jedoch auch einige Wortbedeutungen verloren, wie beispielsweise der Wikipedia Artikel über die Rockband „Queen“, da es bereits mehrere WordNet Synsets zu dem Wort „queen“ gibt. Außerdem können bei einem Vergleich, der ausschließlich auf Lemma-Ebene stattfindet, auch Artikel aus Wikipedia in YAGO aufgenommen werden, welche eine Bedeutung beschreiben, die bereits durch ein WordNet Synset mit von den Artikel Titeln verschiedenen Synonymwörtern abgedeckt sind.

Abgesehen davon gibt es auch Arbeiten, die tatsächlich eine Alignierung von Wortbedeutungen aus zwei Ressourcen vornehmen: Ruiz-Casado *et al.* [2005] alignieren WordNet Synsets und Artikel aus „Simple Wikipedia“ (für Menschen mit eingeschränkten Englischkenntnissen, deutlich kleiner als Wikipedia). Allerdings wird auch hier zunächst eine Zuweisung auf Lemma-Ebene ausgeführt: Alle Artikel deren Titel in genau einem WordNet Synset als Synonymwort vorkommen werden diesen zugewiesen. Falls mehrere Synsets zu einem Artikel gefunden werden, wird ein auf Wortüberlappung basierendes Verfahren (String-basiertes Verfahren) mit Cosinus Distanz genutzt um das dem Artikel ähnlichste Synset zu bestimmen. Dazu wird die Überlappung der im Gloss des Synsets bzw. im ersten Absatz des Wikipedia Artikels verwendeten Wörter gemessen. Eine ähnliche Vorgehensweise haben Ponzetto and Navigli [2010] gewählt, die jedoch WordNet Synsets mit Artikeln aus der vollständigen (nicht vereinfachten) englischen Wikipedia alignieren und mit diesem Verfahren eine erweiterte Ressource mit dem Namen „WordNet++“ erstellen. Das Verfahren zur Berechnung von Ähnlichkeitswerten ist sehr ähnlich zu dem in dieser Arbeit genutzten Ansatz (siehe Abschnitt 5.1.1).

Neben auf Wortüberlappung basierenden Verfahren zur Berechnung von Ähnlichkeitswerten zweier Wortbedeutungen existieren auch semantische Ansätze. Diese Ansätze können besser damit umgehen Ähnlichkeiten zu erkennen, wenn der gleiche Sachverhalt mit Hilfe unterschiedlicher Wörter beschrieben wird. Um die Ähnlichkeit zweier Wörter zu bestimmen, kann beispielsweise die durch semantische Relationen wie Hyponyme und Hyperonyme entstehende Struktur genutzt werden: Je kürzer der Pfad über diese Relationen zwischen zwei Wörtern ist, desto höher ist deren Ähnlichkeit. Ein häufig verwendetes semantisches Verfahren ist der (Personalisierte) PageRank Algorithmus (z.B. Toral *et al.* [2009], Niemann and Gurevych [2011], Meyer and Gurevych [2011]). Semantische Methoden zur Berechnung von Ähnlichkeitswerten erreichen meistens eine höhere Qualität als ausschließlich auf Wortüberlappung basierende Verfahren. Allerdings ist dies in der Regel auch mit einem deutlich höheren Rechenaufwand verbunden.

3.2 Multiple Alignments aus anderen Bereichen

Multiple Alignments von Wortbedeutungen wurden in der Forschung bislang nicht näher betrachtet. Von großer Bedeutung ist das Thema „Multiple Sequence Alignment“ hingegen in der Bioinformatik [Hansen, 2004]. Dort werden Methoden benötigt um Aminosäuresequenzen aneinander auszurichten, das heißt, die Sequenzen sollen an möglichst vielen Stellen übereinstimmen. Die einzelnen Buchstaben in Abbildung 3.1 bezeichnen bestimmte Aminosäuren die bestimmte Ähnlichkeiten zueinander haben. Bei den Alignments können Lücken in den Sequenzen entstehen (sogenannte „Gaps“). Neben paarweisen Zuweisungen muss es auch möglich sein mehr als zwei Sequenzen gleichzeitig einander zuzuweisen, was dann einem Multiplen Alignment entspricht.

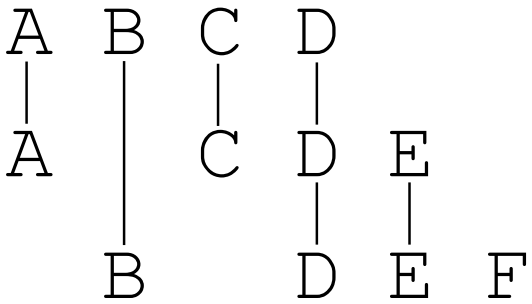


Abbildung 3.1: Multiples Alignment dreier Aminosäuresequenzen

Ein relativ verbreitetes Verfahren zur Berechnung von Multiplen Sequence Alignments ist das progressive Alignment von Feng and Doolittle [1987]. Dabei werden zunächst mit den bekannten Methoden alle Sequenzen paarweise aligniert und ein Ähnlichkeitswert berechnet. Anschließend werden zunächst die beiden Sequenzen mit dem größten Ähnlichkeitswert einander zugewiesen und die Ähnlichkeitswerte zu diesen alignierten Sequenzen neu berechnet (als der Durchschnitt der alten Ähnlichkeitswerte zu den Sequenzen). Dies wird iterativ fortgesetzt bis alle Sequenzen miteinander aligniert sind. Wir werden in Abschnitt 5.1.4 sehen, dass dieses Verfahren dem Hierarchisch Agglomerativen Clustering entspricht, welches wir zur Berechnung multipler Alignments von Wortbedeutungen nutzen werden.

Neben dem „Sequence Alignment“ ist „Ontology Matching“ (oder „Ontology Alignment“) ein des öfteren in der Forschung betrachtetes Thema. Eine Ontologie „typically provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary“[1] [Euzenat and

[1] bietet typischerweise ein Vokabular, das ein Interessensgebiet und eine Spezifikation der Bedeutung von Termen in diesem Vokabular beschreibt

Shvaiko, 2007, S. 1]. Ein Anwendungsbeispiel für eine Ontologie ist beispielsweise die Beschreibung der Waren eines Handelsunternehmens. Will man nun von verschiedenen Handelsunternehmen angebotene Produkte miteinander vergleichen, so muss man die von den Unternehmen verwendeten Ontologien „matchen“: Bei einem Handelsunternehmen findet man ein Buch möglicherweise in der Kategorie „Bücher“, bei einem anderen Unternehmen unter „Literatur“. Eventuell sind zudem einige Informationen (z.B. Maße des Buchs) in einer Ontologie auch gar nicht vorhanden oder mehrere Informationen (z.B. Autor und Titel) unter einem Oberbegriff zusammengefasst. Das Ziel von Ontology Alignment ist folglich übereinstimmende Terme aus verschiedenen Ontologien (wie „Bücher“ und „Literatur“) einander zuzuweisen und dadurch die angebotenen Informationen maschinenlesbar zu machen. Euzenat and Shvaiko [2007] geben einen guten Einblick in das Gebiet „Ontology Matching“.

Ähnlich wie bei dem Alignment von Wortbedeutungen, kann es auch hier sinnvoll sein, mehr als zwei solcher Ontologien in einem multiplen Alignment miteinander zu verknüpfen, beispielsweise um die Waren und Preise von drei Handelsunternehmen zu vergleichen und die angebotenen Informationen zusammentragen zu können. Zhang and Bodenreider [2005] unterscheiden hier zwei mögliche Vorgehensweisen: Entweder je zwei Ontologien werden paarweise aligniert oder eine der Ontologien wird als Referenzontologie ausgewählt, sodass die übrigen Ontologien lediglich auf diese Referenzontologie gemappt werden müssen. Bezüglich dem Alignment von Wortbedeutungen sind beide dieser Ansätze nicht optimal: Die Konstruktion eines multiplen Alignments von Wortbedeutungen aus automatisch berechneten paarweisen Alignments ist sehr fehleranfällig, da Fehler in paarweisen Alignments große Auswirkungen auf das multiple Alignment haben können (siehe Kapitel 4). Bei dem zweiten Ansatz stellt sich die Frage wie mit Wortbedeutungen umgegangen wird, für die keine synonyme Bedeutung in der Referenzressource existiert.

4 Quantitative und Qualitative Datenanalyse

In diesem Kapitel analysieren wir die durch die von UBY angebotenen Senses (UBY Senses) und paarweisen Alignments entstehende Komponentenstruktur und untersuchen inwieweit wir bei der Berechnung multipler Alignments auf vorhandenen paarweisen Alignments aufbauen können. Wir führen sowohl eine quantitative als auch eine qualitative Analyse der gegebenen Daten durch und zeigen verschiedene Probleme auf, die es bei der Entwicklung von Ansätzen zur Berechnung multipler Alignments zu beachten gibt. Dazu stellen wir die von der Ressource angebotenen Senses und paarweisen Alignments in Form eines Graphen dar. Jeder UBY Sense stellt einen Knoten in diesem Graphen dar, die paarweisen Alignments zwischen den Senses werden als Kanten visualisiert. Auf diese Weise erhalten wir einen Graphen mit insgesamt 4.481.321 Knoten und 700.190 Kanten. Dieser Graph ist ungerichtet, ungewichtet und nicht zusammenhängend. Er besteht aus einer Vielzahl von kleineren „Komponenten“. Eine Komponente ist folglich ein zusammenhängender Teilgraph des Gesamtgraphen.

Ziel dieser Arbeit ist es letztlich synonyme Senses zu identifizieren und zu alignieren. Da wir Synonymie als reflexiv, symmetrisch und transitiv ansehen und sich multiple Alignments durch paarweise Alignments darstellen lassen (siehe Abschnitt 2), müssten somit per Definition alle in einer solchen Komponente vorkommenden Senses synonym sein. Tatsächlich muss jedoch beachtet werden, dass die paarweisen Alignments zum größten Teil mit automatischen Methoden generiert wurden, die nicht fehlerfrei arbeiten. Auch wenn die für die paarweisen Alignments bekannten Methoden bereits recht gute Ergebnisse liefern, können einzelne etwas zu hoch oder zu niedrig berechnete Ähnlichkeitswerte (und in Folge dessen ein falsches Alignment) große Auswirkungen auf die Komponenten haben, da ein einzelnes positives Alignment zwei nicht synonyme Komponenten mit allen darin enthaltenen Senses zusammenführt. In dieser Analyse werden wir zeigen inwiefern dies ein Problem für multiple Alignments darstellt.

Im Folgenden werden wir zunächst einige Fehlerindikatoren vorstellen und anschließend anhand dieser Indikatoren in einer quantitativen Datenanalyse untersuchen, wie stark sich Fehler in paarweisen Alignments auf ein multiples Alignment auswirken. Die qualitative Datenanalyse gibt Aufschluss über den Aufbau der Komponenten. In der anschließenden Diskussion werden weitere Probleme und Lösungsansätze vorgestellt.

4.1 Fehlerindikatoren

Die folgenden drei Fehlerindikatoren sind in der Lage festzustellen, ob eine gegebene Komponente Fehler enthält:

Durchmesser

Da alle Senses einer Komponente als synonym angesehen werden, müssten auch alle Senses innerhalb einer Komponente (über eine Kante) miteinander verknüpft sein, sodass die Komponente eine Clique (oder einen vollständigen Graphen [Tittmann, 2003, S. 20]) darstellt. Da der Durchmesser eines Graphen der größte Abstand zweier Knoten im Graphen ist [Tittmann, 2003, S. 34], müsste in diesem Fall jede fehlerfreie Komponente einen Durchmesser von 1 haben. Tatsächlich müssen jedoch einige Besonderheiten berücksichtigt werden: Zu beachten ist, dass nicht für alle Ressourcen-Paare paarweise

Alignments zur Verfügung stehen (insbesondere nicht zwischen Senses der gleichen Ressource). Betrachten wir die paarweisen Alignments zwischen 7 Ressourcen-Paaren, die uns von UBY zur Verfügung gestellt werden (siehe Abbildung 2.3), so beträgt der aus dieser Abbildung ablesbare Durchmesser dieser paarweisen Alignments „P“ = 3. Somit kann keine fehlerfreie Komponente einen Durchmesser „D“ größer als 3 haben. Außerdem sollte der Durchmesser „D“ kleiner sein als die Anzahl der Ressourcen „R“ in der betrachteten Komponente, mit Ausnahme von Komponenten mit 2 oder weniger Ressourcen. Eine Komponente wird folglich dann als korrekt eingestuft, wenn folgende Formel erfüllt ist (andernfalls als fehlerhaft): $D \leq P \wedge (D < R \vee R \leq 2)$. Damit haben wir ein recht einfachen Indikator um Fehler zu identifizieren, da wir lediglich den Durchmesser und die Anzahl der Ressourcen einer Komponente berechnen müssen, um Fehler feststellen zu können.

Es ist allerdings zu beachten, dass in dieser einfachen Form viele Fehler nicht gefunden werden können. Insbesondere fehlende Zyklen können häufig nicht erkannt werden: So kann der Indikator nicht feststellen, dass in Abbildung 4.1 die Kante zwischen dem Sense aus OmegaWiki (deutsch) und dem Sense aus Wikipedia (deutsch) fehlt, da der Durchmesser nicht größer als 3 ist und kleiner als die Anzahl der Ressourcen. Der Durchmesser dieser Komponente bleibt durch Hinzufügen der fehlenden Kante zudem unverändert. Insgesamt können mit dem Indikator 4839 fehlerhafte Komponenten identifiziert werden.

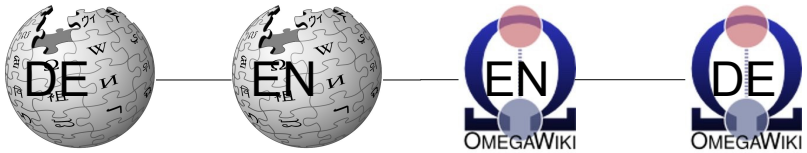


Abbildung 4.1: Fehlerhafte Komponente: Es gibt keine Kante zwischen dem Sense aus OmegaWiki (deutsch) und dem Sense aus Wikipedia (deutsch), obwohl es ein paarweises Alignment zwischen diesen Ressourcen gibt

Gleiche Nachbarn

In einer korrekten Komponente sollten alle Knoten einer Ressource die gleichen Nachbarn haben. Damit ist es bei diesem Indikator nicht notwendig zu prüfen welche paarweisen Alignments vorhanden sind und dies in die Berechnung einzubeziehen. Sobald in einer Komponente von jedem existierenden paarweisen Alignment mindestens eine Kante vorkommt, werden damit alle Fehler gefunden. Sollte andererseits für ein vorhandenes paarweises Alignment gar keine Kante in der Komponente vorkommen, kann dies nicht als Fehler erkannt werden. So würde auch mit diesem Indikator die Komponente aus Abbildung 4.1 als korrekt eingestuft. Insgesamt lassen sich 5755 fehlerhafte Komponenten identifizieren. Darunter wurden 4410 auch vom vorherigen Indikator (Durchmesser) identifiziert. Es gibt jedoch auch 429 Komponenten, bei denen dieser Indikator im Gegensatz zum vorherigen Indikator keine Fehler gefunden hat. Dies betrifft dann Komponenten mit mehr als 4 Ressourcen und großem Durchmesser (siehe Abbildung 4.2).

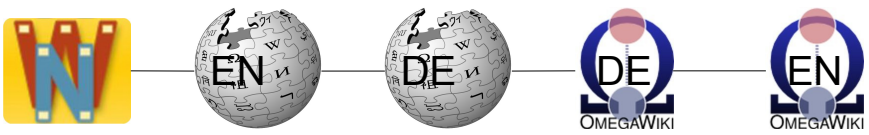


Abbildung 4.2: Fehlerhafte Komponente: Es fehlen die Kanten zwischen den Senses aus OmegaWiki (englisch) und Wikipedia (englisch), sowie zwischen den Senses aus WordNet und OmegaWiki (deutsch)

Fehlende Kanten

Von einer korrekten Struktur innerhalb einer Komponente kann ausgegangen werden, wenn jeder Sense mit allen anderen Senses anderer Ressourcen über eine Kante verbunden ist (siehe Abbildung 4.3). Ein Fehler liegt hingegen vor, wenn innerhalb einer Komponente eine Kante zwischen zwei Senses unterschiedlicher Ressourcen fehlt (siehe Abbildung 4.4).

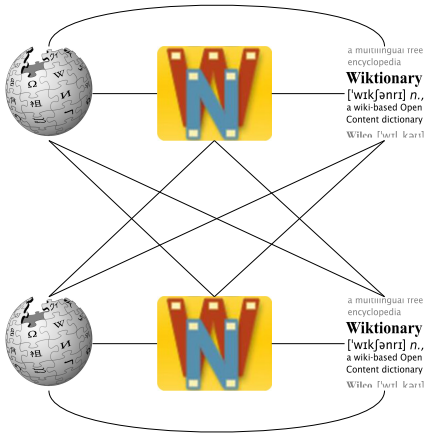


Abbildung 4.3: Korrekte Komponente: Es fehlen keine Kanten

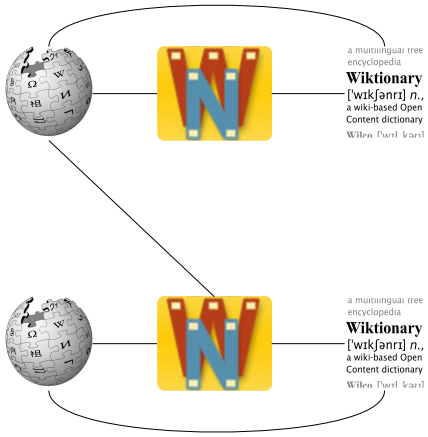


Abbildung 4.4: Fehlerhafte Komponente: Es fehlen 5 Kanten

Da nicht immer paarweise Alignments zwischen sämtlichen Ressourcen-Paaren zur Verfügung stehen, kann man diese Regel dahingehend anpassen, dass Kanten nur dann als fehlend angenommen werden, wenn es zwischen den entsprechenden Ressourcen prinzipiell ein paarweises Alignment gibt. Die Anzahl der fehlenden Kanten berechnet sich dann nach der folgenden Formel, wobei n die Anzahl der Ressourcen darstellt, e die Anzahl der Kanten und r_i (i=1, ..., n) die Anzahl der Knoten von Ressource i in der betrachteten Komponente. $ALIGNED_i$ ist die Menge der Ressourcen, die über ein paarweises Alignment mit Ressource i verbunden sind:

$$(0,5 \cdot \sum_{i=1}^{n} r_i \cdot (\sum_{j \in ALIGNED_i} r_j)) - e$$

In der Komponente aus Abbildung 4.5 berechnen wir die Anzahl an fehlenden Kanten demnach wie folgt: Die Anzahl der Ressourcen n ist 4, die Anzahl der Kanten e ist 11. $r_1 = 1$ (Wikipedia), $r_2 = 1$ (OmegaWiki), $r_3 = 6$ (WordNet), $r_4 = 2$ (Wiktionary). $ALIGNED_1 = \{2,3\}$, $ALIGNED_2 = \{1\}$, $ALIGNED_3 = \{1,4\}$, $ALIGNED_4 = \{3\}$. Die Anzahl der fehlenden Kanten ist somit $0,5 \cdot (1 \cdot (1+6) + 1 \cdot (1) + 6 \cdot (1+2) + 2 \cdot (6)) - 11 = 0,5 \cdot 38 - 11 = 8$. Die entsprechenden fehlenden Kanten sind in Abbildung 4.6 eingezeichnet.

Mit dieser Methode lassen sich 7400 fehlerhafte Komponenten identifizieren (darunter alle von den vorherigen Indikatoren gefundenen Fehler). Trotzdem ist festzustellen, dass wir Fehler nur in Komponenten feststellen können, die mindestens aus drei Ressourcen bestehen (siehe unten) und auch vermeintlich korrekte Komponenten können noch Fehler enthalten.

4.2 Quantitative Datenanalyse

Die Ergebnisse der Analyse der Komponentenstruktur sind Tabelle 4.1 zu entnehmen. Es wird deutlich, dass mit einem Anteil von 86,5% ein Großteil der Senses isoliert ist (d.h. es gibt keine Verbindung dieses Senses/Knotens über ein paarweises Alignment zu einem anderen Sense und die Komponente besteht aus

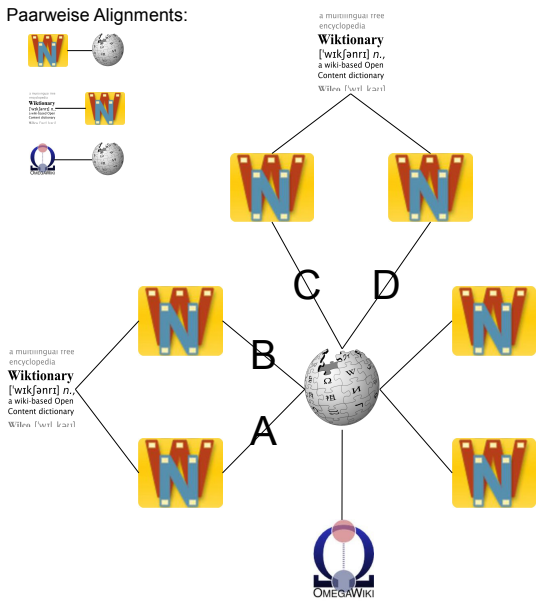


Abbildung 4.5: Eine aus paarweisen Alignments aufgebaute Komponente

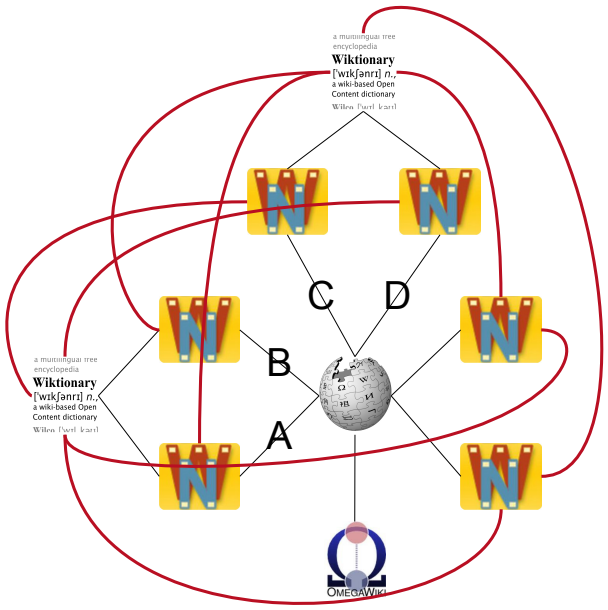


Abbildung 4.6: Es fehlen insgesamt 8 Kanten (rot)

genau einem Knoten). Betrachtet man die Anzahl der Senses aus den verschiedenen Ressourcen, so wird klar, dass der Anteil an isolierten Komponenten recht hoch sein muss: Die Ressource Wikipedia (englisch) enthält 2,9 Millionen Senses. Alle drei Ressourcen mit denen die englische Wikipedia verknüpft ist, enthalten zusammen jedoch nur 1,1 Millionen Senses. Eine genauere Analyse dieser isolierten Senses ist Tabelle 4.2 zu entnehmen. Die hohe Anzahl isolierter Senses ist somit zum Teil den Unterschieden in den Ressourcen zuzurechnen. So enthält Wikipedia beispielsweise sehr viele Informationen über aktuelle Themen (z.B. Personen oder Filme), dafür jedoch ausschließlich Senses der Wortart „Nomen“. Andererseits ist bei der hohen Anzahl an isolierten Komponenten mit einer nicht unerheblichen Menge an False Negatives, also fälschlicherweise nicht einander zugewiesenen Sense-Paaren, in den paarweisen Alignments zu rechnen.

Außerdem stellen wir fest, dass es vereinzelt extrem große Komponenten gibt, was die Annahme aus Abschnitt 2 bestätigt, dass Fehler in paarweisen Alignments große Probleme verursachen können. Die größte gefundene Komponente besteht aus 1654 Knoten, was bei nur 6 Ressourcen unrealistisch ist. Unter den Senses dieser Komponente sind Bedeutungen wie „Head: The human head“ oder „year: A scheduled part of a calendar year spent in a specific activity“, die offensichtlich nicht synonym sind. Nur 0,19% der Komponenten enthalten Fehler entsprechend dem Indikator „Fehlende Kanten“. Dabei sollte jedoch berücksichtigt werden, dass Fehler nur in Komponenten mit mindestens 3 Ressourcen entdeckt werden können. Betrachtet man nur solche Komponenten liegt der Fehleranteil bei 37,23%. Somit wird deutlich, dass es nicht möglich ist durch „Zusammensetzen“ mehrerer paarweiser Alignments ein multiples Alignment zu generieren, da es zu viele Fehler in den paarweisen Alignments gibt und diese sich in einem multiplen Alignment zudem verstärken. Hinzu kommt, dass wir nur innerhalb der zusammenhängenden Komponenten Fehler identifizieren können. Zwei voneinander getrennte (aber eigentlich synonyme) Komponenten werden nicht als Fehler erkannt. Die große Anzahl an sehr kleinen Komponenten deutet wie bereits erwähnt jedoch darauf hin, dass dieser Fall ein häufiges Problem darstellt. Wir beobachten zudem mit steigender Komponentengröße ein massives Ansteigen der Fehlerrate, was naheliegend ist, da Komponenten mit über 20 Senses bei nur 6 Ressourcen eher unwahrscheinlich erscheinen.

#Knoten	#Komponenten	%	#Ressourcen	%Fehler	#Fehlend	Durchmesser
1	3.310.088	86,48	1,0	0,00	-	0,00
2	464.047	12,12	2,0	0,00	-	1,00
3	24.953	0,65	2,2	0,00	-	2,00
4	12.286	0,32	2,7	12,31	1,0	2,26
5	6.272	0,16	2,9	18,37	1,6	2,44
6	3.564	0,09	3,1	31,10	2,2	2,66
7	2.121	0,05	3,3	42,15	3,2	2,87
8	1.234	0,03	3,5	53,73	4,2	3,12
9	786	0,02	3,7	60,81	5,7	3,37
10	574	0,01	3,8	63,59	7,5	3,48
11-20	1364	0,03	4,1	73,46	16,7	4,10
21-30	133	0,00	5,1	95,50	64,4	5,72
31-50	70	0,00	5,5	95,71	180,9	7,41
51-100	24	0,00	5,5	95,83	794,5	9,25
>100	9	0,00	5,8	100,0	63231,7	14,67
Gesamt	3.827.525	100,0	1,1	0,19	86,7	0,15

Tabelle 4.1: UBY-Analyse (#Fehlend = Durchschnittliche Anzahl fehlender Kanten in fehlerhaften Komponenten, # = Anzahl, % = Anteil)

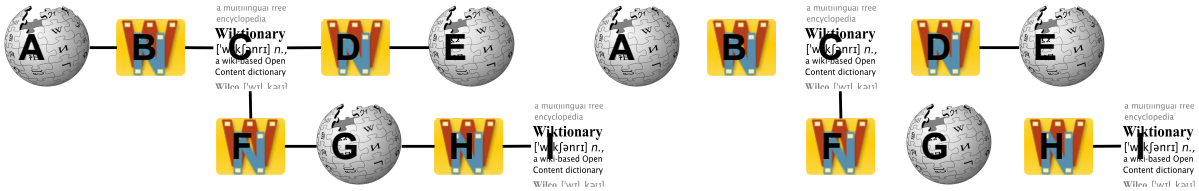
	WordNet	WKT	WP (DE)	WP (EN)	OW (DE)	OW (EN)
# isoliert	79953	371330	381821	2453435	980	22569
% der Senses	38,63	88,02	45,54	83,98	2,82	38,97
% der Komponenten	57,76	88,58	45,55	84,19	4,02	48,08

Tabelle 4.2: Analyse der isolierten Komponenten (WKT = Wiktionary, WP = Wikipedia, OW = OmegaWiki, # = Anzahl, % = Anteil)

4.3 Qualitative Datenanalyse

Die qualitative Datenanalyse zeigt, dass der größere Anteil an Fehlern in den Komponenten durch fälschlicherweise einander zugewiesene Senses entsteht (False Positives). Daraus lässt sich nicht der Schluss ziehen, dass selten Senses fälschlicherweise nicht einander zugewiesen werden (False Negatives), da wir lediglich die vorgegebenen Komponenten untersuchen und somit zwei getrennte, aber eigentlich zusammen gehörende Komponenten nicht als Fehler erkennen können. Ein Beispiel für eine fehlerhafte Komponente mit zahlreichen False Positives ist Abbildung 4.7 (linker Graph) zu entnehmen. Es ist anzunehmen, dass die insgsamt neun Senses deshalb zusammenhängen, weil sie (bis auf Sense B) alle die Bedeutung des Wortes „climber“ beschreiben und dieses Wort zudem häufig in den Beschreibungstexten vorkommt, was dann mehrfach zu hohe Ähnlichkeitswerte zwischen je zwei Senses bewirkt. Bei genauerer Betrachtung der Beschreibungstexte, ist jedoch zu erkennen, dass keineswegs alle dieser Senses synonym sind: Tatsächlich haben wir hier verschiedene Bedeutungen wie die „Kletterpflanze“, den „Sportkletterer“ oder einen „Roboter“. Insgesamt 5 Kanten müssen entfernt werden um die Komponente in eine korrekte Struktur (rechter Graph in Abbildung 4.7) zu überführen.

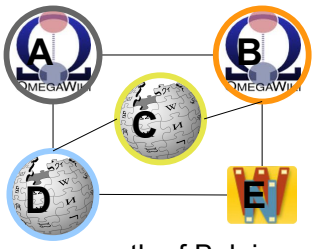
Beispiele wie dieses sind typisch für die aus paarweisen Alignments entstehenden Komponenten: Aufgrund bestimmter in den verschiedenen Beschreibungstexten vorkommender Wörter werden vereinzelt zu hohe Ähnlichkeitswerte berechnet, sodass mehrere unterschiedliche Bedeutungen in einer Komponente zusammenhängen. Wir erkennen jedoch, dass die Komponente nur sehr schwach



A = (Climber): In BEAM robotics, a Climber is a robot that goes upward or downward (...)
B = (mounter): someone who ascends on foot
C = (climber): A person who climbs.
D = (climber): someone seeking social prominence by obsequious behavior
E = (Social climber): A social climber is someone who seeks social prominence, for example by obsequious behavior. (...)
F = (climber): someone who climbs as a sport
G = (Climber): Climber magazine is a British magazine dedicated to all aspects of climbing (...)
H = (climber): a vine or climbing plant that readily grows up a support or over other plants
I = (climber): A plant that climbs, such as a vine.

Abbildung 4.7: Fehlerhafte Komponente (links) und in 6 verschiedene Bedeutungen unterteilte Komponente (rechts) mit Senses aus WordNet, Wikipedia und Wiktionary

zusammenhängend ist: Nicht eine einzige Kante könnte hier entfernt werden ohne die Komponente in zwei Komponenten zu teilen. Dies deutet bereits auf Fehler hin, da in einer korrekten Komponente (wie wir bei Einführung der Fehlerindikatoren erläutert haben) üblicherweise eine stärkere Vernetzung vorliegt. Bei einer stärker vernetzten Komponente (siehe Abbildung 4.8) ist es weniger wahrscheinlich, dass die Komponente aus mehreren unterschiedlichen, zu trennenden, Bedeutungen besteht. Sollte diese Komponente mehrere verschiedene Bedeutungen beinhalten, so müssten hier mindestens zwei Fehler in der Berechnung paarweiser Alignments gemacht worden sein, da mindestens zwei Kanten entfernt werden müssten um die Komponente zu teilen. Zwischen allen fünf Senses dieser Komponente errechnen wir korrekterweise hohe Ähnlichkeitswerte, weil es in den Beschreibungstexten viele übereinstimmende Wörter gibt („country“ bzw. „Land“, „Kingdom“ bzw. „Königreich“, etc.).



A = (The Netherlands): A country in Europe, north of Belgium, officially the Kingdom of the Netherlands. (...)
B = (Königreich der Niederlande): Ein Land, nördlich von Belgien, offiziell das Königreich der Niederlande. (...)
C = (Niederlande): Die Niederlande (Niederländisch: Nederland) sind eine parlamentarische Monarchie und Teil des Königreichs der Niederlande. (...)
D = (Netherlands): The Netherlands is a country in Northwestern Europe, constituting the major portion of the Kingdom of the Netherlands. (...)
E = (Nederland): a constitutional monarchy in western Europe on the North Sea

Abbildung 4.8: Korrekte Komponente mit Senses aus WordNet, Wikipedia (englisch=blau, deutsch=gelb) und OmegaWiki (englisch=grau, deutsch=orange)

Allerdings finden wir eher selten wirklich stark vernetzte Komponenten. Dies hängt damit zusammen, dass zu wenige der genutzten Ressourcen über paarweise Alignments miteinander verknüpft sind. Ein großer Teil der Komponenten ohne identifizierte Fehler, aber mit 3 oder mehr Ressourcen, hat eine Struktur bei der mehrere Knoten einer Ressource mit einem Knoten einer anderen Ressource verknüpft sind. Solche Fälle treten insbesondere durch unterschiedliche Granularitäten der Ressourcen auf. Das heißt eine Ressource beschreibt eine Bedeutung feinkörniger als eine andere Ressource. Die Ressource WordNet scheint beispielsweise deutlich feingranularer zu sein als die Ressource Wikipedia, die dafür eine größere Abdeckung an Bedeutungen aufweist und diese dann allgemeiner beschreibt [Mihalcea, 2007]. Die dadurch entstehenden Probleme werden in Abschnitt 4.4.2 diskutiert. Abgesehen von Granularitätsunterschieden hängen solche Strukturen allerdings auch damit zusammen, dass mehrere Senses der gleichen Ressource zu unterschiedlichen Wörtern, aber mit der gleichen Bedeutung, in einer Komponente vorkommen können. Dieser Punkt wird in Abschnitt 4.4.1 näher beleuchtet.

4.4 Diskussion

Die quantitative und qualitative Datenanalyse hat Erkenntnisse über die Struktur der durch UBY vorgegebenen Komponenten erbracht. Es bestätigt sich, dass die paarweisen Alignments Fehler enthalten und dass diese Fehler große Auswirkungen haben, sofern man die gegebenen Komponenten als multiple Alignments betrachtet. Außerdem sehen wir, dass insbesondere schwach vernetzte Komponenten häufig Fehler enthalten, was wiederum den oben vorgestellten Fehlerindikatoren entspricht.

Im Folgenden gehen wir auf zwei weitere Probleme ein, welche durch die Datenanalyse offengelegt wurden: Dies betrifft einerseits das Problem, dass mehrere Senses der gleichen Ressource und gleichen Bedeutung in einer Komponente vorkommen können und zum anderen Probleme mit unterschiedlichen Granularitäten der Ressourcen.

4.4.1 Sense vs. Synset

Da wir mit Senses arbeiten und nicht mit Synsets, ist zu berücksichtigen, dass des Öfteren mehrere Senses einer Ressource in einer Komponente vorkommen können, die dann zwar zu verschiedenen (Synonym-)Wörtern gehören, aber die gleiche Bedeutung teilen. Dies führt häufig zu Strukturen wie in Abbildung 4.9. Für unsere Arbeit ist dies problematisch, da Bedeutungen, die durch viele Wörter ausgedrückt werden können (z.B. car, auto, automobile, machine, motorcar) auch durch entsprechend viele Senses repräsentiert werden, während Bedeutungen, die durch nur ein Wort ausgedrückt werden können (z.B. chair) nur durch einen einzigen Sense vertreten werden. In einem Graphen führt dies dazu, dass dieser durch Bedeutungen, die aus sehr vielen Senses bestehen, stark aufgebläht wird (siehe Abbildung 4.9). Dieses Ungleichgewicht verursacht Probleme der in Kapitel 5 vorgestellten Ansätze zur Berechnung multipler Alignments, da die Ansätze vergleichsweise viele Kanten entfernen müssen um diese innerhalb einer Ressource synonymen Senses von anderen Senses im Graph abzutrennen.

Für unsere Zwecke ist es daher sinnvoll den Graphen zu „glätten“ indem innerhalb der einzelnen Ressourcen Senses mit gleicher Bedeutung zusammengefasst werden (siehe Abbildung 4.10). Das Ergebnis dieser ressourceninternen Zuweisung sind Synsets. Während es recht unkompliziert ist, Synsets in Senses zu transferieren, ist der umgekehrte Schritt komplizierter, weil die Identifikation synonymer Senses erneut einen (semantischen) Vergleich der Beschreibungstexte erfordert. Um den Aufwand in Grenzen zu halten, haben wir uns im Wesentlichen darauf beschränkt Senses nur innerhalb der Ressourcen WordNet und OmegaWiki zusammenzufassen (bei Wiktionary und Wikipedia nur bei identischen Beschreibungstexten, was nur in Einzelfällen vorkommt) und zudem nur für die in UBY über die paarweisen Alignments zusammenhängenden Komponenten. Das bedeutet, dass einzelne isolierte

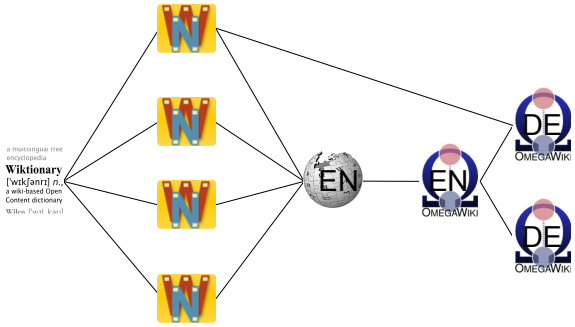


Abbildung 4.9: Aus Senses gebildeter Graph

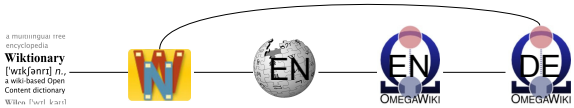


Abbildung 4.10: Aus Synsets gebildeter Graph

Senses nicht zu einem Synset zusammengefasst wurden, was für die spätere Evaluation der Verfahren jedoch keine Auswirkungen hat.

Die Beschränkung auf WordNet und OmegaWiki hängt damit zusammen, dass für diese Ressourcen die Senses in UBY bereits Synsets zugeordnet sind. Somit müssen hier lediglich die sich auf Senses beziehenden paarweisen Alignments auf die Synsets übertragen werden. Es kommt vor, dass beim Zusammenfassen mehrerer Senses einer Ressource zu einem Synset nur eine Teilmenge der Senses mit einem Sense einer anderen Ressource verknüpft ist. Diese Verknüpfung bleibt in dem zusammengesetzten Sense (=Synset) erhalten und gilt dann folglich für alle im Synset enthaltenen Senses. Anzumerken ist auch, dass obwohl sich die in UBY enthaltenen paarweisen Alignments auf Senses beziehen, diese zum Teil zwischen Synsets und Senses berechnet wurden. So wurde beispielsweise ein paarweises Alignment zwischen WordNet Synsets und Wiktionary Senses berechnet. Die Daten müssen daher mit gewisser Vorsicht betrachtet werden, da es dadurch des Öfteren vorkommt, dass beispielsweise ein WordNet Synset mit nur einem Wiktionary Sense aus dem zugehörigen Wiktionary Synset verknüpft ist und nicht mit allen Wiktionary Senses mit der entsprechenden Bedeutung, die dann fälschlicherweise isoliert bleiben.

Bei den Ressourcen Wiktionary und Wikipedia haben wir auf eine automatische interne Zuweisung weitgehend verzichtet, da diese erneut zu Fehlern führt: In Wiktionary sind die Beschreibungen zu Wortbedeutungen verschieden, auch wenn sie die gleiche Bedeutung beschreiben. So teilen die beiden folgenden Wiktionary Senses eine Bedeutung:

- motorcar: an enclosed passenger vehicle powered by an engine.
- automobile: A type of vehicle designed to move on the ground under its own stored power and intended to carry a driver, a small number of additional passengers, and a very limited amount of other load.

Lediglich, wenn die Beschreibungen identisch waren, haben wir hier eine Zuweisung vorgenommen, was nur in wenigen Einzelfällen vorkam. Abgesehen davon, dass die Zusammenfassung aufwendig und fehleranfällig ist, ist bei diesen Ressourcen (Wikipedia und Wiktionary) der Fall, dass mehrere synonyme Senses einer Ressource in einer Komponente auftreten, eher selten. Der Anteil der isolierten Senses unter der Anzahl aller Senses bzw. der Anzahl aller Komponenten der jeweiligen Ressource lässt dies erkennen (siehe Tabelle 4.2): So ist die Differenz dieser (Prozent-)Werte für die Ressourcen Wiktionary und Wikipedia sehr gering, während sie für die Ressourcen WordNet und OmegaWiki recht hoch ist. Der Zusammenhang ist folgendermaßen zu erklären: Wenn die Anzahl der Senses einer Ressource deutlich größer ist als die Anzahl der Komponenten, die mindestens einen Sense dieser Ressource enthalten, dann haben wir im Durchschnitt mehr als einen Sense dieser Ressource in einer Komponente. Dies

lässt jedoch nicht den Schluss zu, dass die Anzahl der Senses, die sich aus einem Synset ergibt, bei Wikipedia und Wiktionary gering ist, sondern lediglich, dass sich diese Senses selten in der gleichen Komponente befinden. So haben wir beispielsweise bei Wiktionary beobachtet, dass häufig einer der synonymen Senses in einer größeren Komponente mit anderen Ressourcen verknüpft ist, während die anderen Senses meist isoliert sind. Dies hängt damit zusammen, dass das paarweise Alignment zwischen WordNet und Wiktionary zwischen WordNet Synsets und Wiktionary Senses berechnet wurde, wobei dann meist nur ein Sense aligniert wurde. Tabelle 4.3 zeigt, dass die ressourceninterne Zusammenfassung von Senses funktioniert: Die Differenz der Prozentwerte für die Ressourcen WordNet und OmegaWiki nimmt im Vergleich zu Tabelle 4.2 stark ab.

	WordNet	WKT	WP (DE)	WP (EN)	OW (DE)	OW (EN)
# isoliert	79953	371330	381821	2453435	980	22569
% der Pseudosynsets	52,67	88,04	45,54	83,98	3,54	47,34
% der Komponenten	57,76	88,58	45,55	84,19	4,02	48,08

Tabelle 4.3: Analyse der isolierten Komponenten nach Bildung von Pseudosynsets (WKT = Wiktionary, WP = Wikipedia, OW = OmegaWiki, # = Anzahl, % = Anteil)

Wir arbeiten bei der Evaluation somit aufgrund der Datengrundlage und des Aufwands nur annähernd mit „Synsets“ und nennen diese daher im folgenden „Pseudesynsets“. Das Ergebnis der Zusammenfassung von Senses zu (Pseudo-)Synsets ist, dass jede Bedeutung einer Ressource genau durch einen Knoten im Graphen repräsentiert wird (siehe Abbildung 4.10).

Die Ergebnisse der Zusammenfassung von Senses zu Pseudosynsets sind Tabelle 4.4 zu entnehmen. Wir verzeichnen, dass der Anteil der kleineren (nicht isolierten) Komponenten nochmals gestiegen ist. Gleichzeitig haben wir nun bereits bei einer Komponentengröße von 5 Knoten eine sehr hohe Fehlerrate (von 18.37% auf 73,95%). Dies ist damit zu erklären, dass nun idealerweise nur noch ein Knoten pro Bedeutung und Ressource in einer Komponente vorkommen sollte. Das heißt die Anzahl der Knoten einer Komponente sollte nicht wesentlich größer sein, als die Anzahl der darin vorkommenden Ressourcen.

#Knoten	#Komponenten	%	#Ressourcen	%Fehler	#Fehlend	Durchmesser
1	3.310.040	86,48	1,0	0,00	-	0,00
2	484.577	12,66	2,0	0,00	-	1,00
3	19.058	0,50	2,4	0,00	-	2,00
4	7.962	0,21	3,4	28,95	1,0	2,73
5	2.472	0,06	3,7	73,95	1,4	3,15
6	1.551	0,04	4,4	92,07	2,2	3,77
7	659	0,02	4,3	93,63	3,8	4,12
8	436	0,01	4,7	96,79	5,7	4,50
9	219	0,01	4,8	97,26	8,4	4,74
10	130	0,00	5,0	95,38	11,1	5,06
11-20	308	0,01	5,1	95,45	25,6	6,11
21-30	32	0,00	5,6	96,88	100,3	8,13
31-50	20	0,00	5,8	95,00	281,9	9,45
51-100	9	0,00	5,8	100,0	846,4	11,22
>100	3	0,00	6,0	100,0	76248,0	21,67
Gesamt	3.827.476	100,0	1,1	0,19	36,8	0,15

Tabelle 4.4: UBY-Analyse nach Pseudosynset Bildung

4.4.2 Unterschiedliche Granularitäten der Ressourcen

Da sich die verwendeten Ressourcen in ihrer Struktur und Abdeckung unterscheiden, muss damit gerechnet werden, dass Senses bzw. Synsets unterschiedlich spezifisch beschrieben und unterteilt werden. Diese Spezifität bezeichnen wir auch als die Granularität einer Ressource. Prinzipiell ist es fast immer möglich eine Bedeutung allgemeiner oder spezieller zu definieren [Ide and Wilks, 2007; Meyer and Gurevych, 2010]. In WordNet gibt es beispielsweise die Synsets „analytical cubism“, „synthetic cubism“ und „cubism“, welche die frühe und die späte Phase des Kubismus, sowie den Kubismus an sich beschreiben. In Wikipedia gibt es hingegen nur einen Artikel „Cubism“, der jedoch auch die frühe und späte Phase einschließt. In einem paarweisen Alignment wäre es folglich eine Option alle drei WordNet Synsets dem Wikipedia Artikel zuweisen. In einem multiplen Alignment ist dies problematischer, weil dadurch automatisch die drei WordNet Synsets zusammengefasst werden, wodurch wir an Spezifität einbüßen (eine Unterscheidung zwischen früher und später Phase des Kubismus ist dann nicht mehr möglich). Hinzu kommt, dass sich solche unterschiedlichen Granularitäten bei mehr als zwei Ressourcen wiederum stärker auswirken. Bei einer derartigen Vorgehensweise orientieren wir uns immer an der Granularität der am wenigsten spezifischen Ressource. Daher ist es sinnvoller in einem solchen Fall nur das WordNet Synset „cubism“ dem Artikel „Cubism“ zuzuweisen und die Verbindung zu den anderen beiden Synsets über WordNet interne Relationen (z.B. Hyperonyme) herzustellen.

Die Analyse der UBY Komponenten nach Zusammenfassung der Senses zu Pseudosynsets zeigt, dass unterschiedliche Granularitäten durchaus vorkommen. Wir erkennen dies daran, dass die Anzahl der Ressourcen nicht gleichermaßen mit der Anzahl der Knoten in einer Komponente steigt (siehe Tabelle 4.4), was bedeutet, dass hier mehrere Pseudosynsets der gleichen Ressource in einer Komponente zusammengefasst werden. Eine wirkliche Lösung für das Problem gestaltet sich schwierig, da wir hier mit dem Begriff Synonymie arbeiten, aber zugleich nur selten eine 100%ige Übereinstimmung zweier Senses oder Synsets vorfinden. Wir entscheiden uns im Folgenden dafür aus den oben genannten Gründen nur in wenigen Fällen (sehr starke Ähnlichkeitswerte) die Zuweisung mehrerer Pseudosynsets der gleichen Ressource zuzulassen.

In Abbildung 4.11 sehen wir ein aus paarweisen Alignments (aus UBY) zusammengesetztes Beispiel bei dem die Ressource WordNet die Bedeutung „Lense“ bzw. „Linse“ aus Wikipedia in insgesamt 5 Bedeutungen unterteilt und somit deutlich feinkörniger ist. In WordNet unterscheiden wir konkave und konvexe Linsen, sowie Kameralinsen, Linsensysteme und die optische Linse allgemein, während wir in Wikipedia nur einen einzigen Artikel haben, der jedoch alle diese Bedeutungen abdeckt. Wie schon in dem Beispiel oben empfiehlt es sich hier lediglich den allgemeinen WordNet Eintrag zu der optischen Linse zuzuweisen. Ein Blick in die semantischen Relationen von WordNet offenbart, dass die anderen Bedeutungen (konvexe/konkave Linse usw.) Hyponyme der allgemeinen Bedeutung sind, sodass die Verbindung zu diesen spezielleren Bedeutungen erhalten bleibt. Als Ergänzung zur qualitativen Datenanalyse fügen wir hinzu, dass insbesondere derartige Konstruktionen (mehrere Knoten einer Ressource verknüpft mit einem Knoten einer anderen Ressource) von den vorgestellten Fehlerindikatoren nicht als Fehler erkannt werden.

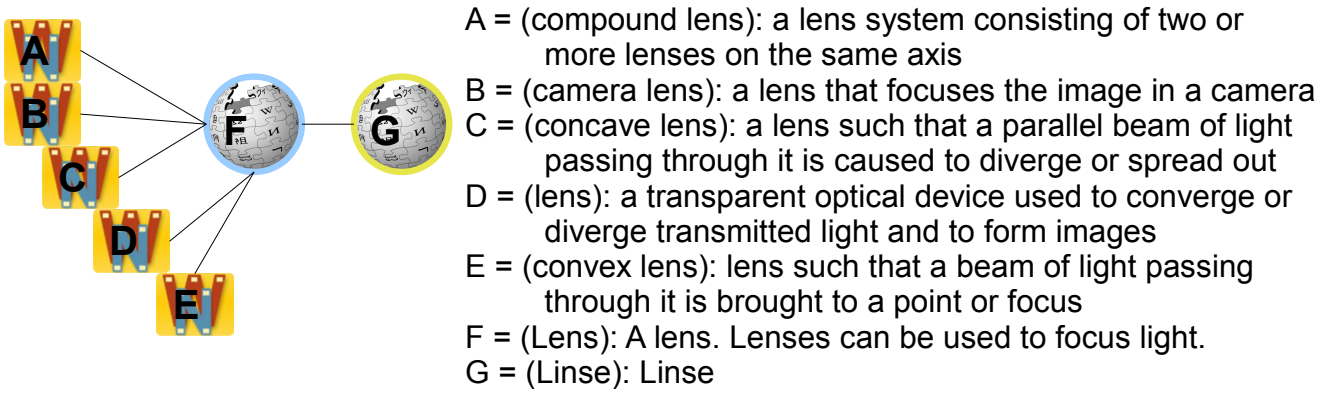


Abbildung 4.11: Unterschiedliche Granularitäten von WordNet und Wikipedia (englisch=blau, deutsch=gelb)

5 Ansätze

Bei der Berechnung multipler Alignments definieren wir einen konstruktiven und einen korrektiven Ansatz. Beide Ansätze werden im Folgenden im Detail beschrieben. Anschließend stellen wir die beiden Ansätze einander gegenüber.

5.1 Konstruktiver Ansatz

Ausgangspunkt des konstruktiven Ansatzes ist eine Menge von Pseudosynsets aus einer beliebigen Anzahl ($\geq$3) verschiedener lexikalischer Ressourcen. Das bedeutet, dass keinerlei Informationen aus zuvor berechneten paarweisen Alignments benötigt werden. Im ersten Schritt werden paarweise Ähnlichkeitswerte zwischen sämtlichen Paaren an Pseudosynsets unterschiedlicher Ressourcen berechnet (siehe Abschnitt 5.1.1). Die Ähnlichkeit zwischen zwei Pseudosynsets der gleichen Ressource setzen wir gleich 0, da wir es nach Möglichkeit vermeiden wollen mehrere Pseudosynsets der gleichen Ressource zusammenzufassen (siehe Abschnitt 4.4.2). Das Ganze lässt sich wiederum als Graph visualisieren, bei dem die Knoten den Pseudosynsets entsprechen und die gewichteten Kanten die Ähnlichkeit zwischen je zwei Pseudosynsets angeben.

Das Ziel ist es aus diesem Graphen ein multiples Alignment zu erzeugen, indem wir den Graphen in einzelne stark zusammenhängende Komponenten bzw. Cluster unterteilen. Jede dieser Komponenten repräsentiert dann idealerweise eine aus synonymen Pseudosynsets verschiedener Ressourcen zusammengesetzte Bedeutung. Um den erhofften Qualitätsvorteil gegenüber paarweisen Alignments zu erreichen sind hierzu Algorithmen notwendig, die die globale Struktur des Graphen ausnutzen. Diesbezüglich eignen sich insbesondere Clustering-Algorithmen von denen wir einige in Abschnitt 5.1.4 vorstellen. Daneben stellen wir zudem noch das Maß „Topological Overlap“ vor (siehe Abschnitt 5.1.5), das unter anderem dazu genutzt werden kann die Gewichte eines Graphen entsprechend seiner Topologie anzupassen. Dieses Maß können wir folglich nutzen, um zu hohe bzw. zu niedrige paarweise Ähnlichkeitswerte anhand der globalen Struktur nach unten bzw. nach oben zu korrigieren.

Das in den vorherigen Kapiteln beschriebene Problem, dass einzelne Ähnlichkeitswerte fälschlicherweise zu hoch oder zu niedrig sind, führt - anders als bei paarweisen Alignments - in einem multiplen Alignment nicht zwangsläufig zu einem falschen Alignment, da entsprechende Ähnlichkeitswerte durch andere Ähnlichkeitswerte ausgeglichen werden können.

Da der Schwerpunkt dieser Arbeit darin liegt, die Vorteile multipler Alignments gegenüber paarweisen Alignments aufzuzeigen, beschäftigen wir uns hier nicht näher mit der Wahl geeigneter paarweiser Ähnlichkeitsmaße (siehe hierzu Kapitel 3), sondern arbeiten mit einem Standardverfahren, das wir im Folgenden vorstellen werden. Neben diesem Standardverfahren zur Berechnung paarweiser Ähnlichkeitswerte existieren zahlreiche andere Verfahren, die in Vorarbeiten zu paarweisen Alignments betrachtet wurden: Mihalcea *et al.* [2006] geben eine Übersicht über einige Maße zur Berechnung der Ähnlichkeit von Texten. Desweiteren gibt es semantische Methoden wie den Personalisierten PageRank Algorithmus (z.B. Toral *et al.* [2009], Niemann and Gurevych [2011]). Diese können besser damit umgehen, wenn beispielsweise unterschiedliche Wörter zur Beschreibung der gleichen Bedeutung genutzt werden (siehe Kapitel 3). Außerdem werden wir auf einige Herausforderungen, wie die hohe Komplexität des Ansatzes sowie die Vergleichbarkeit von Ähnlichkeitswerten eingehen. Anschließend

stellen wir verschiedene Clustering-Algorithmen sowie das Maß „Topological Overlap“ zur Erstellung des multiplen Alignments vor.

5.1.1 Berechnung von Ähnlichkeitswerten

Für die Berechnung von paarweisen Ähnlichkeitswerten verwenden wir einen, in der Literatur häufig genutzten, „bag of words“-Ansatz mit Cosinus Ähnlichkeit (siehe z.B. Ruiz-Casado *et al.* [2005]). Dazu erstellen wir zunächst für alle Pseudosynsets eine Bag of Words. Diese enthält alle Wörter aus dem jeweiligen Beschreibungstext, sowie die Synonymwörter zu dem Pseudosynset, wobei diese zunächst in ihre Grundform überführt werden. Außerdem werden Stopwords entfernt (unter Stopwords sind Wörter zu verstehen, die keinen Einfluss auf die Bedeutung eines Texts haben, z.B. Konjunktionen oder Präpositionen). Da wir neben vier englischsprachigen Ressourcen auch zwei deutschsprachige Ressourcen nutzen, übersetzen wir die Beschreibungstexte dieser beiden Ressourcen mit Microsoft Translator[1] zuvor ins Englische, um die Wörter miteinander vergleichen zu können.

Zur Berechnung der Ähnlichkeit zwischen zwei Pseudosynsets wird für die beiden zugehörigen Bags of Words je ein Vektor erstellt, der für alle Wörter, die in einer der Bags vorkommen, den Wert 0 oder 1 enthält, je nachdem, ob das entsprechende Wort in der zugehörigen Bag vorkommt oder nicht (siehe Abbildung 5.2). Mit der Cosinus-Distanz berechnen wir anschließend die Übereinstimmung der beiden Vektoren in Form eines Ähnlichkeitswerts im Intervall [0, 1]. Die Cosinus-Distanz beschreibt den Winkel zwischen zwei Vektoren (siehe Abbildung 5.1) und ist folgendermaßen definiert:

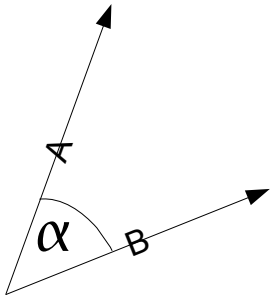


Abbildung 5.1: Cosinus Distanz

$$cosDist(A,B) = \cos(\alpha) = \frac{\sum_i A_i \cdot B_i}{\sum_i A_i^2 \cdot \sum_i B_i^2}$$

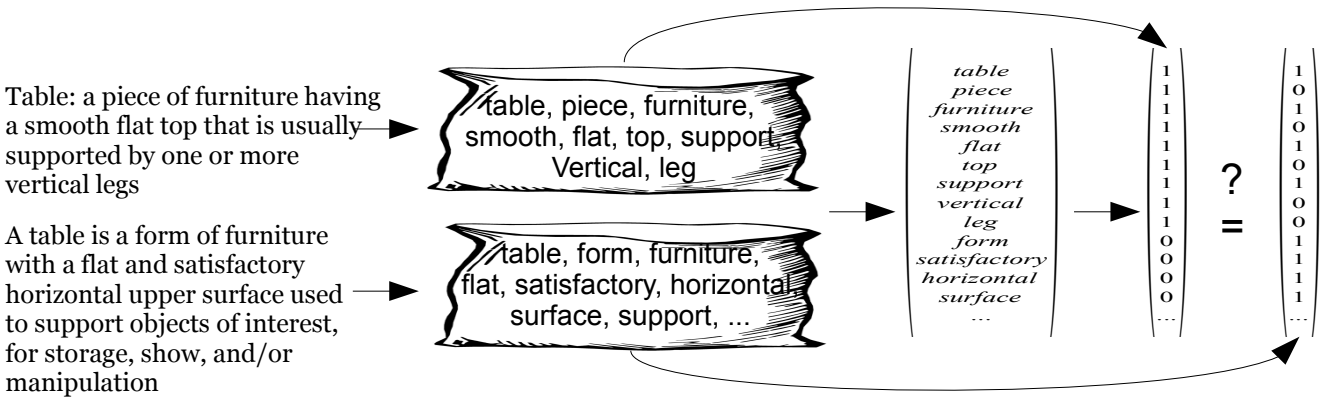


Abbildung 5.2: Erstellung von Bags of Words und zugehöriger Vektoren aus zwei Beschreibungstexten

[1] http://www.microsofttranslator.com/dev/

5.1.2 Komplexität des Ansatzes

Der konstruktive Ansatz ist mit einem hohen Aufwand verbunden. Gäbe r_i die Anzahl der Senses von Ressource i an und sei n die Anzahl der Ressourcen, dann müssen theoretisch $\sum_{i=1}^{n-1} r_i(\sum_{j=i+1}^{n} r_j)$ Ähnlichkeitswerte berechnet werden. Wie bei paarweisen Alignments auch, ist es jedoch nicht notwendig, Ähnlichkeitswerte zwischen sämtlichen Sense Paaren unterschiedlicher Ressourcen zu berechnen. Um den Aufwand zu reduzieren berechnet man für jeden Sense eine geringe Anzahl an Kandidaten aus anderen Ressourcen und berechnet lediglich zu diesen Kandidaten Ähnlichkeitswerte. Damit reduziert sich bei c Kandidaten pro Sense die Anzahl der zu berechnenden Ähnlichkeitswerte auf maximal $c(n-1)\sum_{i=1}^{n} r_i$.

Allerdings ist auch die Berechnung von Kandidaten für alle Ressourcen kein einfach zu lösendes Problem. Es ist sehr wichtig, dass, sofern es ein synonymes Pseudosynset in der jeweils anderen Ressource gibt, dieses sich unter den Kandidaten befindet. So reicht es beispielsweise nicht aus, zu dem WordNet Pseudosynset „disk cache“ nach Wikipedia Artikeln mit dem gleichen Titel zu suchen, da der zugehörige Artikel den Titel „Page cache“ hat. Erschwert wird die Kandidatenextraktion zudem durch die Multilingualität (deutsche und englische Ressourcen), welche eine automatische und wahrscheinlich nicht fehlerfreie Übersetzung notwendig macht. Wolf and Gurevych [2010] haben eine Methode entwickelt, um Wikipedia Artikel als Kandidaten für WordNet Synsets zu finden. Ähnliche Verfahren sind für alle Kombinationen der verwendeten Ressourcen zu entwickeln.

Berechnet man die Kandidaten jeweils nur in eine Richtung (z.B. für jeden WordNet Sense die Kandidaten aus Wikipedia, aber nicht andersherum), so ist im günstigsten Fall die Berechnung von $c\sum_{i=1}^{n-1} r_i(n-i)$ Ähnlichkeitswerten notwendig. Idealerweise sollte die Kandidatenauswahl symmetrisch sein, sodass dies keinen Qualitätsverlust bedeutet: Wenn also ein Sense A aus WordNet die Kandidaten C und D aus Wikipedia extrahiert und Sense B aus WordNet die Kandidaten D und E aus Wikipedia, dann sollte, sofern man die Kandidaten für Wikipedia bestimmt, Sense D aus Wikipedia die Kandidaten A und B aus WordNet extrahieren (siehe Abbildung 5.3). Um hier eine möglichst kleine Anzahl an zu berechnenden Ähnlichkeitswerten zu erhalten, sollte man mit der kleinsten Ressource beginnen. Die Größe der Ressourcen unterscheidet sich zum Teil sehr deutlich. So ist die englische Wikipedia mit 2,9 Millionen Senses deutlich größer als beispielsweise das deutsche OmegaWiki mit knapp 28.000 Senses. Die durchschnittliche Kandidatenanzahl pro Sense bereits existierender Gold-Standards für paarweise Alignments (zwischen WordNet und Wikipedia bzw. WordNet und Wiktionary) beträgt etwa c=5. Für die n=6 UBY Ressourcen aus Tabelle 5.1 ist somit die Berechnung von $5 \cdot (27708 \cdot 5 + 47674 \cdot 4 + 151815 \cdot 3 + 421795 \cdot 2 + 838427 \cdot 1) = 12{,}3$ Millionen Ähnlichkeitswerten notwendig.

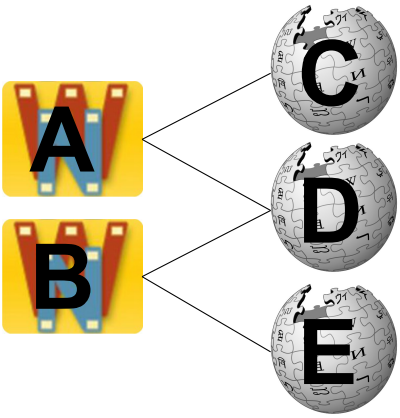


Abbildung 5.3: Symmetrische Kandidatenauswahl

WN	WktEN	WikiDE	WikiEN	OwDE	OwEN
151.815	421.795	838.427	2.921.454	27.708	47.674

Tabelle 5.1: Anzahl der Senses verschiedener UBY Ressourcen

Die Vorverarbeitung der Daten (Erstellung von Bags of Words) erfordert bei einer Prozessortaktung von 2 GHz etwa 100 Millisekunden pro Sense bzw. Pseudosynset, was bei einer Anzahl von 4 Millionen etwa 5 Tage in Anspruch nimmt. Die Berechnung der Ähnlichkeitswerte mit dem recht einfachen hier verwendeten String-basierten Ansatz (siehe Abschnitt 5.1.1) erfordert einen Zeitaufwand von etwa 20 Millisekunden pro Ähnlichkeitswert. Bei 12,3 Millionen Ähnlichkeitswerten bedeutet dies einen Zeitaufwand von etwa 3 Tagen. Bei semantischen Verfahren zur Berechnung von Ähnlichkeitswerten ist jedoch mit einem deutlich höheren Zeitaufwand zu rechnen. Mit dem Personalisierten PageRank Verfahren würde die Berechnung derart vieler Ähnlichkeitswerte mehr als einen Monat in Anspruch nehmen [Agirre and Soroa, 2009].

Wir stellen folglich fest, dass die Kandidatenauswahl von großer Bedeutung ist. Ohne Kandidatenauswahl müssten wir für unsere 6 Ressourcen mehrere Billionen Ähnlichkeitswerte berechnen, was selbst mit dem einfachen String-basierten Verfahren unmöglich ist. In der Evaluation nutzen wir die durch die Gold-Standards (auf denen wir die Evaluation durchführen) vorgegebenen Kandidaten (siehe Abschnitt 6.2), sodass die automatische Berechnung von Kandidaten nicht erforderlich ist.

5.1.3 Normalisierung von Ähnlichkeitswerten

Eine weitere Herausforderung bei dem konstruktiven Ansatz ist die Vergleichbarkeit von Ähnlichkeitswerten. So muss ein Ähnlichkeitswert zwischen zwei Pseudosynsets aus WordNet und Wiktionary von 0,7 keine größere Ähnlichkeit bedeuten als ein Ähnlichkeitswert von 0,6 zwischen zwei Pseudosynsets aus WordNet und Wikipedia. Dies hängt mit den ressourcenspezifischen Beschreibungen der Pseudosynsets zusammen: Wikipedia hat beispielsweise eher längere Beschreibungstexte, während WordNet und Wiktionary Pseudosynsets meistens nur mit wenigen Wörtern beschrieben werden. Um die Ähnlichkeitswerte dennoch miteinander vergleichen zu können, ist folglich eine Normalisierung notwendig.

Das Ziel dieser Normalisierung ist es, die Ähnlichkeitswerte so anzupassen, dass ein größerer Ähnlichkeitswert in jedem Fall auch eine größere Ähnlichkeit bedeutet und das unabhängig von den Ressourcen, zwischen denen die Ähnlichkeiten berechnet wurden. In dem Beispiel oben würde man folglich alle Ähnlichkeitswerte zwischen Pseudosynsets der Ressourcen WordNet und Wiktionary etwas reduzieren und alle Ähnlichkeitswerte zwischen Pseudosynsets der Ressourcen WordNet und Wikipedia etwas erhöhen, sodass eine größere Ähnlichkeit auch durch einen größeren Ähnlichkeitswert ausgedrückt wird.

Dazu haben wir für 1000 zufällig ausgewählte Paare von Pseudosynsets zwischen sämtlichen Ressourcen die Ähnlichkeit berechnet und daraus den Mittelwert gebildet, wobei Paare von Pseudosynsets mit dem Ähnlichkeitswert 0 bei Bildung des Mittelwerts ignoriert wurden, da sonst die Anzahl der synonymen Paare zu großen Einfluss auf die Normalisierung hat. Wikipedia hat beispielsweise sehr viele Pseudosynsets, während OmegaWiki nur vergleichsweise wenige Pseudosynsets hat. Somit ist hier mit einer deutlich größeren Anzahl an Samples mit dem Ähnlichkeitswert 0 zu rechnen. Die durchschnittlichen Ähnlichkeitswerte für die verschiedenen Ressourcenpaare sind Tabelle 5.2 zu entnehmen. Es fällt auf, dass immer, wenn die Ressource Wikipedia beteiligt ist, der Mittelwert besonders niedrig ist. Dies ist (wie bereits erwähnt) damit zu erklären, dass die Ressource Wikipedia im Vergleich zu allen anderen Ressourcen deutlich längere Beschreibungstexte hat.

Für die Normalisierung benötigen wir eine streng monoton steigende Funktion, die durch die Punkte (0, 0), (1, 1), sowie durch den Punkt (avg, 0.5) geht, wobei „avg“ für den zuvor berechneten durchschnittlichen Ähnlichkeitswert des jeweiligen Ressourcenpaares steht. Wir normalisieren die

Ähnlichkeitswerte folglich so, dass der durchschnittliche Ähnlichkeitswert in normalisierter Form den Wert 0,5 erhält. Die zugehörige Funktion sieht folgendermaßen aus: $f(x) := log_2(x^a + 1)$. Dabei ist für „x“ der zu normalisierende Ähnlichkeitswert einzusetzen und „a“ ist so zu wählen, dass die Funktion für den zuvor berechneten Mittelwert („avg“) den Wert 0,5 annimmt. Abbildung 5.4 zeigt beispielhaft zwei Normalisierungsfunktionen mit a=0,5 bzw. a=0,2. Das heißt der zugehörige Mittelwert beträgt 0,012 bzw. 0,172. Diese beiden Werte haben in normalisierter Form somit die gleiche Ähnlichkeit von 0,5.

Ressource A	Ressource B	Mittelwert	a-Wert für Funktion
WordNet	Wikipedia (en)	0.0675	0.3270
WordNet	Wiktionary	0.1673	0.4930
WordNet	Wikipedia (de)	0.0889	0.3641
WordNet	OmegaWiki (de)	0.1080	0.3960
WordNet	OmegaWiki (en)	0.1336	0.4379
WordNet	WordNet	0.1566	0.4754
Wiktionary	Wiktionary	0.1530	0.4695
Wiktionary	Wikipedia (en)	0.0665	0.3251
Wiktionary	Wikipedia (de)	0.0702	0.3317
Wiktionary	OmegaWiki (de)	0.1436	0.4542
Wiktionary	OmegaWiki (en)	0.1196	0.4151
Wikipedia (en)	Wikipedia (en)	0.0607	0.3145
Wikipedia (en)	Wikipedia (de)	0.0628	0.3184
Wikipedia (en)	OmegaWiki (en)	0.0720	0.3350
Wikipedia (en)	OmegaWiki (de)	0.0730	0.3368
Wikipedia (de)	Wikipedia (de)	0.0700	0.3315
Wikipedia (de)	OmegaWiki (de)	0.0807	0.3501
Wikipedia (de)	OmegaWiki (en)	0.0694	0.3304
OmegaWiki (de)	OmegaWiki (de)	0.4188	1.0125
OmegaWiki (de)	OmegaWiki (en)	0.2706	0.6743
OmegaWiki (en)	OmegaWiki (en)	0.4507	1.1060

Tabelle 5.2: Normalisierung von Ähnlichkeitswerten

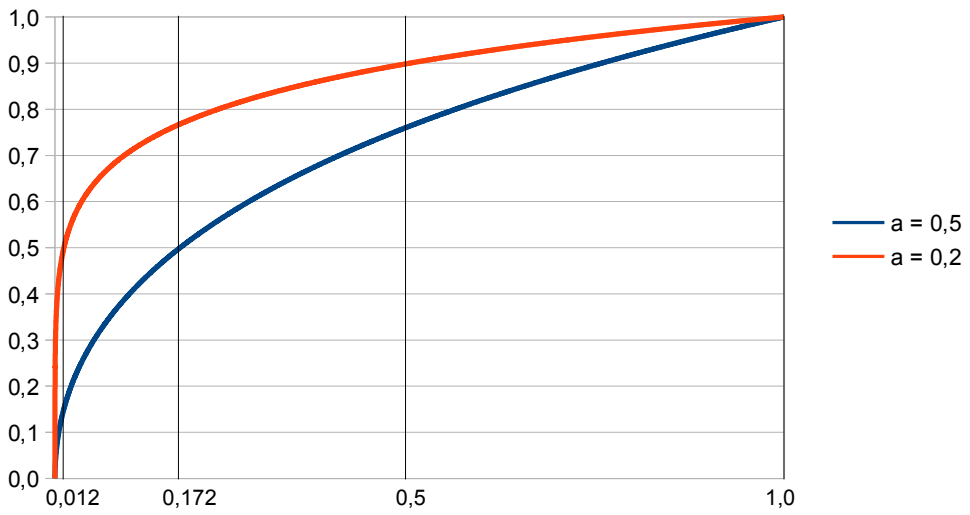


Abbildung 5.4: Normalisierungsfunktionen für Ähnlichkeitswerte im Vergleich

Es ist zu berücksichtigen, dass in die Berechnung des durchschnittlichen Ähnlichkeitswertes zwischen zwei Ressourcen meistens nur etwa 50 Werte eingeflossen sind, obwohl wir Ähnlichkeitswerte für 1000 Samples berechnet haben. Dies hängt damit zusammen, dass wir alle Samples mit einem Ähnlichkeitswert von 0 ignoriert haben, da dies das Ergebnis verfälscht hätte (siehe oben). Um die Qualität der Normalisierung zu verbessern, könnte eine größere Anzahl an Samples betrachtet werden. Für unsere Zwecke halten wir die gewählte Vorgehensweise jedoch für ausreichend.

5.1.4 Clustering-Algorithmen

Die Clustering-Algorithmen haben die Aufgabe den mit normalisierten paarweisen Ähnlichkeitswerten gewichteten Graphen in einzelne Cluster zu zerteilen. Alle Knoten eines Clusters gelten dann als synonym und werden in dem multiplen Alignment einander zugewiesen.

Hierarchisch Agglomeratives Clustering

Beim Hierarchisch Agglomerativen Clustering [King, 1967; Jain *et al.*, 1999] sehen wir am Anfang jeden Knoten des ungerichteten, gewichteten Graphen als ein Cluster an. Wir fassen dann iterativ die beiden Cluster mit der größten Ähnlichkeit zu einem Cluster zusammen und berechnen die Ähnlichkeitswerte dieses zusammengefassten Clusters zu den anderen Clustern neu. Dies wiederholen wir iterativ bis wir entweder eine bestimmte Anzahl an Clustern unterschreiten oder der größte Ähnlichkeitswert zwischen zwei Clustern unterhalb eines Schwellenwerts liegt.

Abbildung 5.5 zeigt einen sehr einfachen Beispielgraphen: In der ersten Iteration werden hier die Cluster A und B zusammengefasst, da sie mit einem Ähnlichkeitswert von 0,9 die größte Ähnlichkeit aufweisen. Für die Neuberechnung der Ähnlichkeitswerte sind die drei Methoden „single-link“, „complete-link“ und „average-link“ bekannt. Bei „single-link“ wird für die Ähnlichkeit zwischen einem Cluster C und dem aus A und B zusammengesetzten Cluster der kleinste Ähnlichkeitswert zwischen C und A bzw. B übernommen. Bei „complete-link“ wird analog der größte Ähnlichkeitswert übernommen. Die für unsere Problemstellung wohl am besten geeignete Methode ist jedoch „average-link“. Hier wird der Durchschnitt der beiden Kanten gebildet. Für den Fall, dass eine der beiden Kanten fehlt, ist für diese Kante der Ähnlichkeitswert 0 anzunehmen. Die Methode hat den Vorteil, dass (wie bei paarweisen Alignments) nicht länger nur ein Kantengewicht über eine Zuweisung entscheidet. So kann beispielsweise eine Kante, die sehr deutlich über dem Schwellenwert liegt, eine Kante, die knapp unter dem Schwellenwert liegt, ausgleichen. Dadurch können Fehler, die in einem paarweisen Alignment entstehen, vermieden werden. Bei einem Schwellenwert von 0,5 würden wir das Clustering mit average-link in Abbildung 5.5 nach der ersten Iteration bereits beenden und hätten dann als Ergebnis zwei Cluster (A/B und C). Alle Knoten innerhalb eines solchen Clusters gelten dann als synonym und sind in einem multiplen Alignment einander zugewiesen.

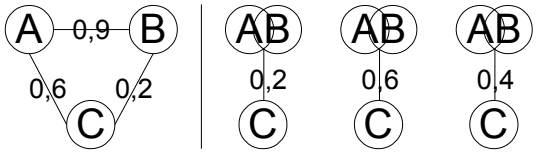


Abbildung 5.5: links: Beispielgraph, rechts: 1. Iteration mit „single-link“, „complete-link“ und „average-link“ (von links nach rechts)

Ein wesentlicher Vorteil dieses Clustering-Algorithmus besteht darin, dass er leicht verständlich und nachvollziehbar ist und auch auf großen Graphen effizient arbeitet. Außerdem müssen wir die Anzahl der gewünschten Cluster nicht im Voraus festlegen, sondern können als Abbruchkriterium den gleichen Schwellenwert nutzen wie bei paarweisen Alignments. So können sowohl größere als auch minimale Cluster (ein einzelner Knoten) entstehen. Diese Eigenschaft ist für die Berechnung multipler Alignments von großer Bedeutung, da wir sowohl Bedeutungen haben können, die nur in einer der Ressourcen vorkommen (minimales Cluster), als auch Bedeutungen, die in allen Ressourcen vorkommen.

Newman Clustering

Das von Newman and Girvan [2004] entwickelte Clustering-Verfahren startet mit dem gesamten Graphen als einzigem Cluster und zerteilt diesen Graphen dann iterativ. Das Verfahren ist somit „divisiv“ und nicht „agglomerativ“ wie der vorherige Ansatz. Ursprünglich wurde das Verfahren für ungewichtete Graphen vorgestellt, kann jedoch leicht auf gewichtete Graphen ausgeweitet werden [Newman, 2004].

Der Algorithmus basiert auf sogenannten „Betweenness Measures“. Ein Cluster ist in der Regel durch eine starke Vernetzung mit Kanten innerhalb des Clusters charakterisiert, während zwischen zwei verschiedenen Clustern eher wenige Kanten vorhanden sind. Dadurch müssen alle Pfade zwischen zwei Knoten aus verschiedenen Clustern über diese wenigen Kanten verlaufen, die dadurch einen hohen Betweenness-Wert erhalten. Ein Beispiel für ein Betweenness Measure ist „Shortest Paths“ (kürzeste Wege): Zu jedem Knoten im Graph werden dabei die kürzesten Wege zu allen anderen Knoten berechnet und für jede Kante gezählt, wie häufig sie in diesen kürzesten Wegen vorkommt. In Abbildung 5.6 sehen wir links einen Beispielgraphen (die Zahlen geben die Betweenness-Werte an) und rechts die kürzesten Wege für die einzelnen Knoten. Wir stellen fest, dass ausschließlich die senkrechte Kante in der Mitte in allen kürzesten Wegen vorkommt. Dadurch erhält diese Kante einen hohen Betweenness-Wert.

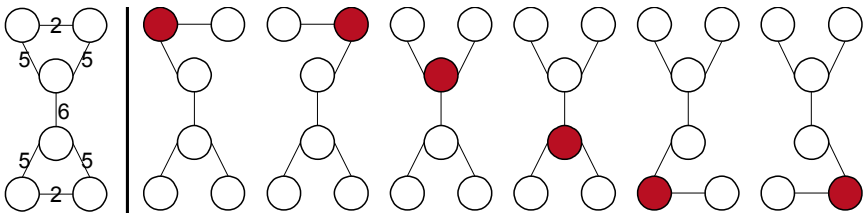


Abbildung 5.6: links: Beispielgraph, rechts: Berechnung der kürzesten Wege ausgehend von dem jeweils markierten Knoten

Newman and Girvan [2004] empfehlen für die meisten Probleme das Shortest Paths Maß, weshalb auch wir uns für dieses Maß entschieden haben. Darüber hinaus stellen sie noch weitere Betweenness Measures vor: Bei „Random Walk“ läuft ein Läufer zufällig über den Graphen. Kanten, die dabei häufig durchlaufen werden, erhalten einen hohen Betweenness-Wert. Bei dem Maß „Current Flow“ wird der Fluss zwischen verschiedenen Quellen und Senken gemessen (verschiedene Knoten-Paare werden hierfür ausgewählt) und Kanten mit einem hohen Fluss erhalten einen hohen Betweenness-Wert.

Der Newman Algorithmus berechnet folglich im ersten Schritt den Betweenness-Wert für sämtliche Kanten eines Graphen und entfernt dann die Kante mit dem höchsten Betweenness-Wert. Dies wird iterativ wiederholt bis der Graph durch das Entfernen von Kanten in eine bestimmte Anzahl an nicht zusammenhängenden Komponenten zerteilt wurde oder bis der maximale Betweenness-Wert unterhalb eines Schwellenwerts liegt. Der Algorithmus kann sehr einfach auf gewichtete Graphen übertragen

werden indem die auf dem ungewichteten Graphen berechneten Betweenness-Werte für jede Kante durch deren Gewicht geteilt werden [Newman, 2004].

Der Nachteil dieser Methode (im Vergleich zum vorherigen Ansatz) ist die deutlich höhere Komplexität. Wir nutzen für die Berechnung der Betweenness-Werte das Java Universal Network/Graph (JUNG) Framework[2]. Der dort verwendete Algorithmus nutzt das „Shortest Paths“ Maß und basiert auf dem Algorithmus von Brandes [2001]. Die Laufzeit des Algorithmus zur Berechnung der Betweenness-Werte ist mit $O(n^2 + nm)$ angegeben, wobei n die Anzahl der Knoten und m die Anzahl der Kanten darstellt.

Da wir die Betweenness-Werte in jeder Iteration und für jede Kante neu berechnen müssen, steigt der Aufwand bei großen Graphen (>1000 Kanten) stark an. Wir haben deshalb bei entsprechenden Graphen in jeder Iteration mehrere Kanten auf einmal entfernt bevor wir die Betweenness-Werte neu berechnet haben. Da wir bei noch sehr großen Graphen ohnehin sehr viele Kanten entfernen müssen, bis es es zu einer Teilung des Graphen kommt und in den meisten Fällen die in einer Iteration entfernten Kanten auch bei mehreren Iterationen entfernt worden wären, sollte dadurch die Qualität nicht deutlich beeinträchtigt werden. In jedem Fall gewinnen wir dadurch an Effektivität. Wenn wir 10 Kanten pro Iteration entfernen, so sind wir dadurch etwa 10 mal so schnell wie wenn wir nur eine Kante pro Iteration entfernen.

PageRank Clustering

Ein weiterer Clustering-Algorithmus wurde von Avrachenkov *et al.* [2008] vorgestellt und basiert auf dem PageRank Algorithmus [Page *et al.*, 1999]. Dieser bewertet die Knoten eines Graphen anhand dessen ein- und ausgehenden Kanten. Bildlich gesprochen erhält am Anfang jeder Knoten ein Initialgewicht, von dem er in mehreren Iterationen einen Teil über seine ausgehenden Kanten an seine Nachbarknoten abgibt. Gleichzeitig erhält er über die eingehenden Kanten Gewicht von seinen Nachbarknoten. Ein Knoten mit vielen eingehenden Kanten bzw. mit eingehenden Kanten eines Knotens mit sehr hohem Gewicht erhält dadurch selbst ein hohes Gewicht und somit einen hohen PageRank-Wert. Der PageRank Algorithmus bewertet folglich die strukturelle Bedeutung von Knoten.

Im ersten Schritt werden mit dem PageRank Algorithmus die Knoten des Graphen bewertet und so mögliche Clusterzentren bestimmt. Nach einer Überprüfung, ob die gefundenen Clusterzentren sich in unterschiedlichen Clustern befinden, werden nun die restlichen Knoten des Graphen diesen Clusterzentren zugeordnet. Dazu wird für jedes Clusterzentrum ein Personalisierter PageRank Algorithmus [Haveliwala, 2003] ausgeführt. In diesem Personalisierten PageRank Algorithmus erhält das jeweilige Clusterzentrum das volle Initialgewicht, die anderen Knoten bekommen dagegen kein Initialgewicht. In den Iterationen des Personalisierten PageRank Algorithmus wird das Initialgewicht des Clusterzentrums dann in dem Graphen entsprechend der gewichteten Kanten verteilt. Ein Knoten wird letztlich dem Cluster zugeordnet, für den es den höchsten PageRank-Wert der Durchläufe des Personalisierten PageRank Algorithmus erhalten hat.

Große Graphen lassen sich mit diesem Algorithmus in einer Iteration in eine beliebige Anzahl an vorher festgelegten Clustern teilen. Ein Nachteil dieses Clustering-Algorithmus im Vergleich zu den zuvor vorgestellten Verfahren ist jedoch, dass es somit auch keinen Schwellenwert gibt, wodurch es schwierig ist zu entscheiden an welcher Stelle man das Clustering beendet. Eine mögliche Abbruchbedingung besteht darin, die Graphen solange zu teilen, bis deren Größe (gemessen an der Anzahl an Knoten) unter einem bestimmten Wert liegt. Einige einfache Experimente, bei denen wir die Komponenten in Cluster von weniger als 6 Knoten zerlegt haben, zeigen, dass das PageRank Clustering etwas schlechtere

[2] http://jung.sourceforge.net/

Ergebnisse liefert als das Newman Clustering. Im Allgemeinen ist es wenig sinnvoll die Größe von Clustern als Abbruchbedingung heranzuziehen, da viele Pseudosynsets nur in ein oder zwei Ressourcen vorkommen. Eine aus beispielsweise 4 Knoten bestehende Komponente würde daher (auch bei geringen Ähnlichkeitswerten) nicht weiter unterteilt. Wir haben dieses Verfahren aus diesem Grund nicht näher evaluiert.

Label Propagation

Bei diesem Verfahren [Raghavan *et al.*, 2007] wird anfangs jeder Knoten als eigenes Cluster angesehen, wobei jeder Knoten durch eine eindeutige Bezeichnung („Label“) identifiziert wird. Im Folgenden werden nun alle Knoten des Graphen in zufälliger Reihenfolge durchgegangen, wobei jeder Knoten das Label annimmt, welches die Mehrheit seiner Nachbarn trägt. Bei gewichteten Graphen (wie wir sie haben) übernehmen wir für den Knoten das Label mit der größten Summe an Kantengewichten zwischen dem Knoten und seinen Nachbarn. Dies wiederholen wir bis sich die Labels nicht mehr verändern.

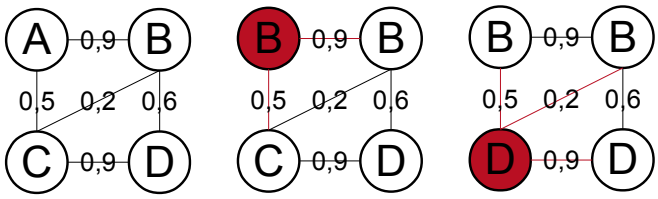


Abbildung 5.7: Label Propagation: Die Knoten werden in der Reihenfolge A, C, D, B durchlaufen (bei D und B keine Veränderung, daher nicht abgebildet)

Die Knoten des Graphen in Abbildung 5.7 werden in der (zufälligen) Reihenfolge A, C, D, B durchlaufen. Der Knoten A nimmt somit das Label von Knoten B an, weil die Kante eine Ähnlichkeit von 0,9 aufweist, was größer ist als 0,5 zu Label C. Im zweiten Schritt nimmt Knoten C das Label D an, weil 0,9 größer ist als 0,7 (0,5 + 0,2). Bei den anderen beiden Knoten verändert sich das Label nicht mehr, sodass der Algorithmus an dieser Stelle terminiert. Wir haben den Graph folglich in 2 Cluster (B und D) geteilt.

Eine für unsere Zwecke sehr negativ zu bewertende Eigenschaft des Verfahrens ist, dass der Algorithmus niemals Cluster produziert, die aus nur einem einzigen Knoten bestehen. Da wir für unseren Task jedoch ein Verfahren benötigen, das sowohl sehr kleine als auch etwas größere Cluster produzieren kann, müssen wir feststellen, dass dieses Verfahren hierfür nicht geeignet ist. Diese Eigenschaft ist für uns deshalb von großer Bedeutung, weil es (wie wir in Kapitel 4 gesehen haben) zahlreiche Pseudosynsets gibt, die nur in einer einzigen Ressource vorkommen. Selbst bei dem Beispiel aus Abbildung 5.8 erhält Knoten D mit diesem Algorithmus jedoch das gleiche Label wie die Knoten A, B und C, obwohl die Ähnlichkeit sehr gering ist. Das Verfahren wird im Folgenden daher nicht weiter betrachtet.

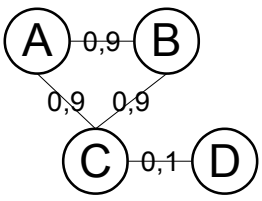


Abbildung 5.8: Trotz geringer Ähnlichkeit zwischen C und D erhalten alle Knoten das gleiche Label

5.1.5 Topological Overlap

Das von Ravasz *et al.* [2002] vorgestellte Maß „Topological Overlap“ können wir nutzen, um den Kanten eines Graphen Ähnlichkeitswerte entsprechend dessen Topologie zuzuweisen. Die gewichtete Variante des Verfahrens von Li and Horvath [2007] ermöglicht es uns, vor der Anwendung anderer Algorithmen, wie den oben beschriebenen Clustering-Algorithmen, möglicherweise fehlerhafte Ähnlichkeitswerte entsprechend der Topologie des Graphen nach oben oder nach unten zu korrigieren und die Verfahren dadurch zu verbessern.

Das Maß funktioniert so, dass der Ähnlichkeitswert zweier Knoten, die gemeinsame Nachbarn haben, zu denen sie wiederum hohe Ähnlichkeitswerte haben, nach oben korrigiert wird, während der Ähnlichkeitswert von zwei Knoten ohne gemeinsame Nachbarn oder mit geringen Ähnlichkeitswerten zu diesen nach unten korrigiert wird. Der Topological Overlap t_{ij} (der neu berechnete Ähnlichkeitswert) für die Kante zwischen den Knoten i und j berechnet sich nach folgender Formel von Li and Horvath [2007], wobei a_{ij} für den Ähnlichkeitswert zwischen Knoten i und j steht:

$$t_{ij} = \frac{\sum\limits_{u \neq i,j} a_{iu} \cdot a_{ju} + a_{ij}}{min(\sum\limits_{u \neq i} a_{iu} - a_{ij}; \sum\limits_{u \neq j} a_{ju} - a_{ij}) + 1}$$

Abbildung 5.9 zeigt einen Beispielgraphen (links), bei dem wir leicht erkennen können, dass hier zwei Cluster (A/B/C und D/E/F) vorliegen. Wir erkennen jedoch, dass die Ähnlichkeitswerte der in dem Graphen rot hervorgehobenen Kanten von dieser Intuition abweichen und vermutlich zu hoch bzw. zu niedrig sind. Durch Anwenden des Topological Overlap wird das Gewicht der Kante B/C erhöht, da die beiden Knoten B und C hohe Ähnlichkeitswerte zu ihrem gemeinsamen Nachbarknoten A haben. Das Gewicht der Kante A/D wird hingegen nach unten korrigiert, da diese beiden Knoten keinen gemeinsamen Nachbarknoten besitzen. Auch die Gewichte der anderen Kanten verändern sich: So verringert sich beispielsweise das Gewicht der Kante A/C, da C eine geringe Ähnlichkeit zu dem gemeinsamen Nachbarn B hat, was (wie wir festgestellt haben) ein Fehler ist. Allerdings sind die Änderungen hier nicht so stark. Bei mehr gemeinsamen Nachbarn (ohne Fehler in den Ähnlichkeitswerten) würde sich der eine Fehler zudem noch geringer auswirken. Auf dem resultierenden Graphen mit den korrigierten Gewichten kann nun eines der oben beschriebenen Clustering-Verfahren ausgeführt werden, um den Graphen zu teilen.

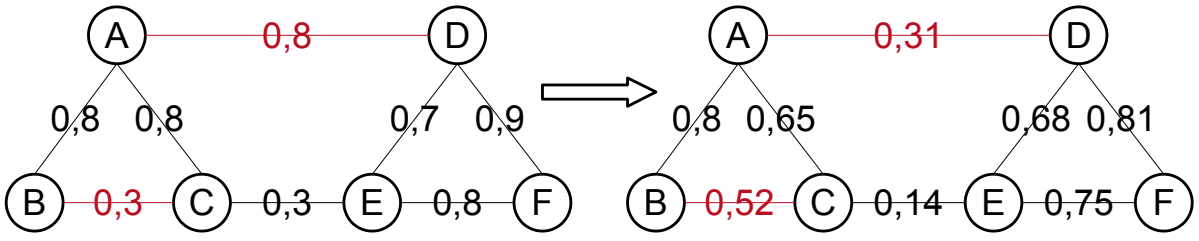


Abbildung 5.9: Veränderung der Gewichte eines Graphen durch Topological Overlap

5.2 Korrektiver Ansatz

Der korrektive Ansatz baut im Gegensatz zum konstruktiven Ansatz direkt auf paarweisen Alignments auf. Dazu wird aus sämtlichen vorhandenen paarweisen Alignments zunächst ein ungerichteter, ungewichteter und nicht zusammenhängender Graph aufgebaut. Dies entspricht der Vorgehensweise aus Kapitel 4, wo wir bereits einen entsprechenden Graphen aus den von UBY zur Verfügung gestellten Daten generiert haben. Im Unterschied zum konstruktiven Ansatz, bei dem wir Ähnlichkeitswerte zwischen allen Paaren an Pseudosynsets berechnen und die Kanten des Graphen damit gewichten, verwenden wir hier nur die binäre Unterscheidung zwischen Zuweisung (Kante) oder keine Zuweisung (keine Kante) und erhalten somit einen ungewichteten Graphen. Die Gewichtung der Zuweisungen mit den in den paarweisen Alignments berechneten Ähnlichkeitswerten ist zwar möglich, der hier vorgestellte korrektive Ansatz hat jedoch den Vorteil, dass er bis zu einem gewissen Grad auch ohne diese Ähnlichkeitswerte auskommt.

Wie in Kapitel 4 beschrieben können Fehler in paarweisen Alignments zu extrem großen und fehlerhaften Komponenten führen. Während dieses Problem beim konstruktiven Ansatz (siehe Abschnitt 5.1) behoben wird, indem einzelne zu hohe oder zu niedrige Ähnlichkeitswerte durch andere Ähnlichkeitswerte ausgeglichen werden können, wird beim korrektiven Ansatz auf den einzelnen Komponenten des ungewichteten Graphen eine Fehlerkorrektur durchgeführt, indem fehlende Kanten hinzugefügt oder die Komponenten durch das Entfernen von Kanten geteilt werden.

Die einzelnen durch die paarweisen Alignments gebildeten zusammenhängenden Komponenten werden im ersten Schritt auf mögliche Fehler untersucht. Diese Analyse basiert zunächst ausschließlich auf der Struktur der Komponenten und erfordert keine Ähnlichkeitswerte. Wir nutzen dafür den Fehlerindikator „Fehlende Kanten“, den wir bereits in Abschnitt 4.1 eingeführt haben. In den Abbildungen 5.10 und 5.11 sehen wir je ein Beispiel für eine von dem Fehlerindikator als korrekt bzw. fehlerhaft eingestufte Komponente. Die Knoten entsprechen jeweils einem Pseudosynset aus der durch das Ressourcenlogo gekennzeichneten Ressource. Die in Abbildung 5.11 fehlenden 5 Kanten sind in Abbildung 5.10 vorhandenen, weshalb diese Ressource korrekt eingestuft wird.

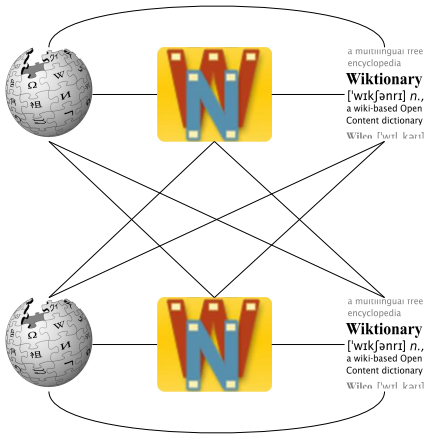


Abbildung 5.10: Korrekte Komponente: Es fehlen keine Kanten

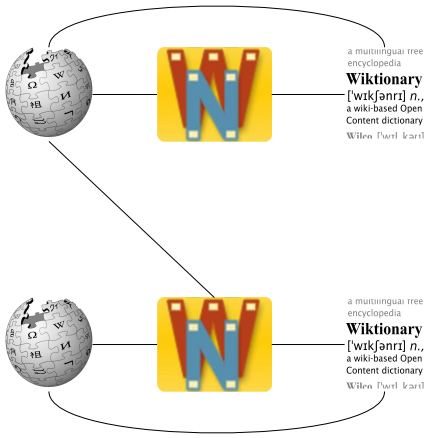


Abbildung 5.11: Fehlerhafte Komponente: Es fehlen 5 Kanten

Hat man Fehler in einer Komponente identifiziert, so versucht man im nächsten Schritt diese Fehler zu korrigieren und die Komponente in eine Struktur zu überführen, in der keine Kanten fehlen. Dies erreicht man, indem entweder fehlende Kanten hinzugefügt werden oder Kanten entfernt werden, so dass die Komponente in mehrere Komponenten zerteilt wird. In dem Beispiel aus Abbildung 5.11 fehlen

insgesamt fünf Kanten (vergleiche Abbildung 5.10). Andererseits müsste man nur eine Kante entfernen (die senkrechte Kante zwischen Wikipedia und WordNet) um die Komponente in zwei Komponenten ohne fehlende Kanten zu zerteilen. Es ist davon auszugehen, dass die Veränderung von nur einer Kante (= Kante entfernen) zu einem besseren Ergebnis führt, als die Veränderung von fünf Kanten (Kanten hinzufügen), weshalb wir uns in diesem Beispiel dafür entscheiden, die Komponente durch das Entfernen der einen Kante zu teilen. Um zu bestimmen welche Kanten entfernt werden sollten, eignen sich wiederum Clustering-Algorithmen (siehe Abschnitt 5.2.1).

Letztlich muss eine Abwägung stattfinden, ob Kanten hinzugefügt oder entfernt werden. Jede dieser Änderungen in einer Komponente bezeichnen wir im Folgenden als „Edit-Operation“. Die naheliegendste Vorgehensweise ist es, das Hinzufügen und Entfernen von Kanten als gleich zu bewerten und somit anzustreben die Anzahl an Edit-Operationen zu minimieren. Der Komponente aus Abbildung 5.12 fehlen 8 Kanten, die in Abbildung 5.13 eingezeichnet sind. Wir haben nun mehrere Optionen: Option A besteht darin die 8 fehlenden Kanten hinzuzufügen und die Komponente somit nicht zu teilen (8 Edit-Operationen, im Folgenden „No-Split Option“). Eine weitere Möglichkeit ist es die Komponente durch das Entfernen von Kanten zunächst in zwei Komponenten zu teilen und dann innerhalb dieser beiden Komponenten gegebenenfalls noch fehlende Kanten hinzuzufügen. Dazu berechnen wir uns mit einem Clustering-Algorithmus (siehe Abschnitt 5.2.1) eine oder mehrere Möglichkeit(en) (im Folgenden „Split Option“). Die No-Split Option ist folglich eine Menge von Edit-Operationen, bei denen ausschließlich Kanten hinzugefügt werden, während eine Split Option eine Menge von Edit-Operationen ist, bei der sowohl Kanten entfernt werden (Teilen der Komponente), als auch Kanten hinzugefügt werden können (innerhalb der geteilten Komponenten fehlende Kanten).

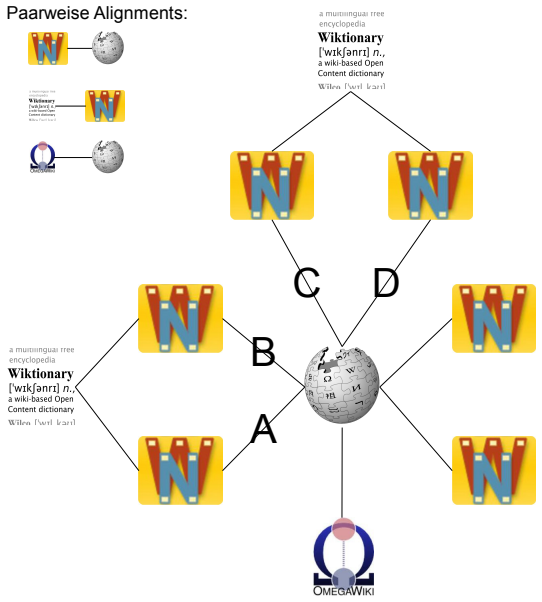


Abbildung 5.12: Eine aus paarweisen Alignments aufgebaute Komponente

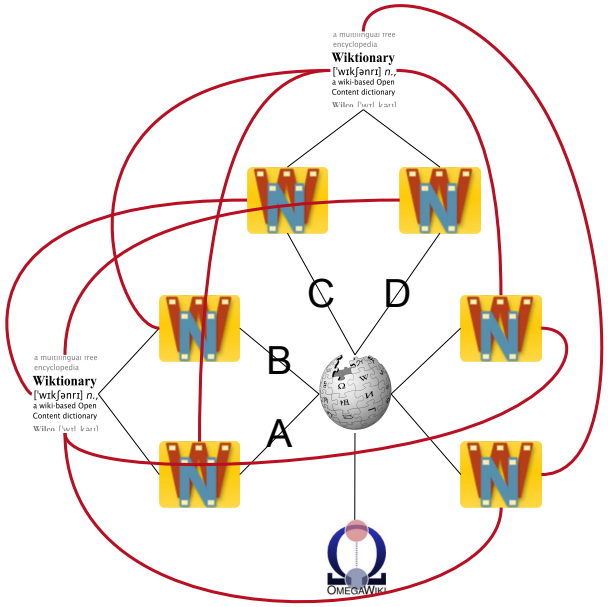


Abbildung 5.13: Es fehlen insgesamt 8 Kanten (rot)

Nehmen wir an, der Algorithmus liefert uns die folgenden beiden Split Optionen: Entfernen der Kanten A und B bzw. Entfernen der Kanten C und D. Beide dieser Split Optionen erfordern 4 Edit-Operationen, weil jeweils zwei Kanten entfernt werden und den aus den Splits resultierenden Komponenten jeweils noch 2 Kanten fehlen. Da die Anzahl der Edit-Operationen der beiden Split Optionen mit 4 besser ist als bei der No-Split Option (8 Edit-Operationen), müssen wir uns folglich zwischen diesen beiden

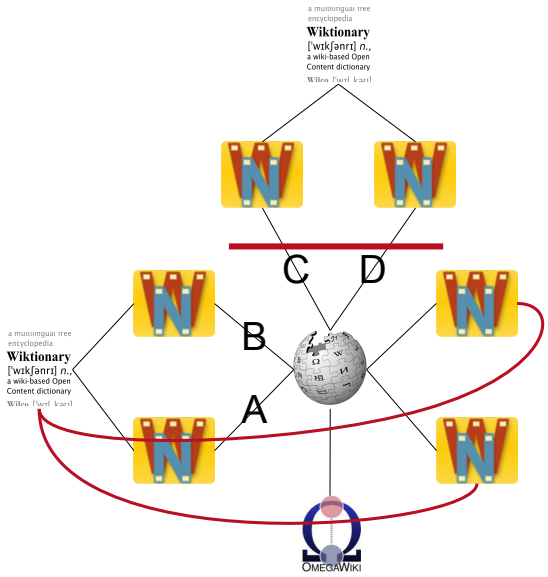


Abbildung 5.14: Zerteilen der Komponente: 2 Kanten entfernt, 2 Kanten fehlen (rot)

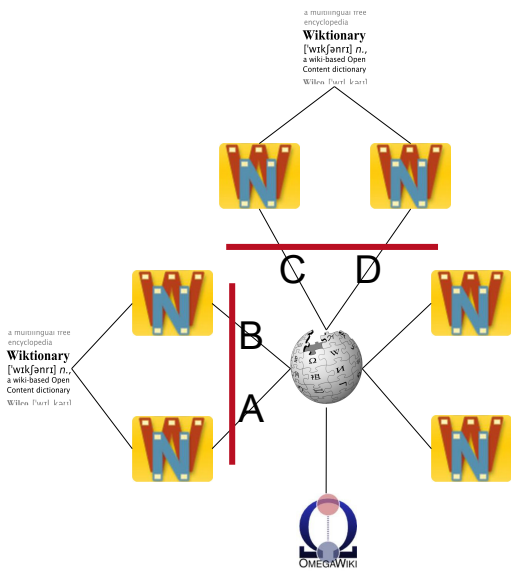


Abbildung 5.15: Erneutes Zerteilen: 4 Kanten entfernt, keine fehlenden Kanten

Split Optionen entscheiden. Die einfachste Lösung für das Problem ist es zufällig eine der Optionen zu wählen. Erfolgsversprechender ist es jedoch für alle Kanten, die für ein Entfernen in Frage kommen (hier: A, B, C, D), Ähnlichkeitswerte zu berechnen und dann die Kanten mit der geringsten „Summe entfernter Ähnlichkeitswerte“ zu entfernen. Hätten die relevanten Kanten in Abbildung 5.14 also die Ähnlichkeitswerte A=0.8, B=0.8, C=0.7, D=0.8, so würde die Komponente durch Entfernen der Kanten C und D zerteilt, da deren Summe entfernter Ähnlichkeitswerte mit 1,5 kleiner ist als der entsprechende Wert der Kanten A und B (1,6). Die Idee dahinter ist, dass wir bei gleicher Anzahl an Edit-Operationen eher Kanten mit einem geringen Ähnlichkeitswert entfernen als Kanten mit hohen Ähnlichkeitswerten.

In der nächsten Iteration müssen wir uns erneut entscheiden die beiden noch fehlenden Kanten zu ergänzen (Option A, No-Split Option, Edit-Operationen = 2) oder die Komponente durch Entfernen der Kanten A und B nochmals zu teilen (Option B, Split Option, Edit-Operationen = 2). Da die Summe entfernter Ähnlichkeitswerte der No-Split Option jedoch grundsätzlich 0 ist, würde die No-Split Option in einem solchen Fall immer vorgezogen werden. Eine Möglichkeit damit umzugehen ist es der No-Split Option ebenfalls eine Summe entfernter Ähnlichkeitswerte >0 zuzuweisen (etwa im Bereich des für paarweise Alignments trainierten Schwellenwerts). Analog wird bei dieser Vorgehensweise eine Split Option mit einer entfernten Kante in den meisten Fällen auch einer Split Option mit zwei entfernten Kanten vorgezogen (bei gleicher Anzahl an Edit-Operationen). Hier wäre es daher denkbar die Summe entfernter Ähnlichkeitswerte durch die Anzahl der entfernten Kanten zu teilen. Wir haben verschiedene Vorgehensweisen in der Evaluation des Verfahrens getestet (siehe Abschnitt 6.3.2).

Der korrektive Ansatz erlaubt eine Fehlerkorrektur ausschließlich innerhalb der durch die paarweisen Alignments vorgegebenen Komponenten. Dies hat den folgenden Nachteil: Wenn durch einen Fehler in den paarweisen Alignments zwei Komponenten mit der gleichen Bedeutung fälschlicherweise nicht zusammenhängen, dann können diese beiden Komponenten durch den korrektiven Ansatz auch nicht zusammengeführt werden. Zwar werden durch den korrektiven Ansatz nicht nur Kanten entfernt, sondern auch ergänzt, allerdings nur innerhalb bereits verbundener Komponenten. Abbildung 5.16 zeigt ein entsprechendes Beispiel: Während Kante A durch den korrektiven Ansatz ergänzt werden kann, ist dies mit Kante B nicht möglich. Es ist jedoch damit zu rechnen, dass die Anzahl dieser Fehler mit

steigender Anzahl an verfügbarer paarweiser Alignments zwischen verschiedenen Ressourcen abnimmt. Abgesehen davon können nur Fehler korrigiert werden, die von unserem Fehlerindikator „Fehlende Kanten“ als solche identifiziert werden. So können beispielsweise nur in Komponenten mit 3 oder mehr Ressourcen überhaupt Fehler festgestellt werden, da sonst keine globale Struktur vorhanden ist. Positiv anzumerken ist beim korrektiven Ansatz hingegen der vergleichsweise geringe Aufwand: Wir müssen nur wenige Ähnlichkeitswerte berechnen und arbeiten auf relativ kleinen Graphen (vorgegeben durch die paarweisen Alignments).

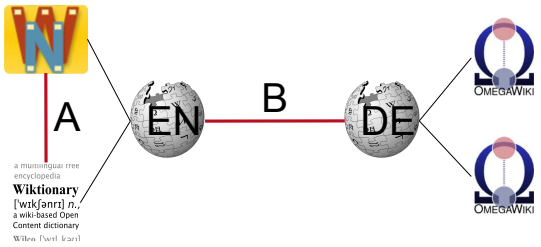


Abbildung 5.16: Nur Kanten innerhalb verbundener Komponenten können hinzugefügt werden: Kante A kann ergänzt werden, Kante B nicht

5.2.1 Algorithmus zum Finden von Split Optionen

Der konstruktive Ansatz benötigt Clustering-Algorithmen zur Ermittlung von Split Optionen. Geeignet für diesen Zweck ist beispielsweise das bereits in Abschnitt 5.1.4 beschriebene Newman Clustering (für ungewichtete Graphen). Dieser Algorithmus entfernt iterativ Kanten mit hohen Betweenness-Werten was zu einer Teilung des entsprechenden Graphen und somit zu einer Split Option führt. Da wir immer nur die Kante mit dem höchsten Betweenness-Wert entfernen, erhalten wir in der Regel nur eine einzige Split Option, was bei noch recht großen Komponenten jedoch in der Regel ausreichend ist.

Eine andere Vorgehensweise besteht darin sich alle möglichen Split Optionen eines Graphen zu berechnen. Dadurch ist es möglich aus einer Vielzahl an Optionen eine Wahl zu treffen und es besteht nicht die Gefahr, dass eventuell korrekte Split Optionen bei der Entscheidung für eine der Optionen gar nicht berücksichtigt werden, weil der Clustering-Algorithmus diese nicht ausgewählt hat. Dieser Fall ist aus folgendem Grund möglich: Das Newman Clustering betrachtet ausschließlich Topologie und Kantengewichte, bezieht jedoch nicht ein, zu welcher Ressource ein Knoten gehört und arbeitet folglich anders als der von uns genutzte Fehlerindikator.

Daher haben wir uns den folgenden Algorithmus überlegt, dessen Pseudocode Abbildung 5.17 zu entnehmen ist. Im Wesentlichen probiert der Algorithmus durch das Entfernen von Kanten alle Möglichkeiten aus, um den Graphen in zwei Komponenten zu teilen und gibt am Ende alle minimalen Lösungen zurück. Eine Split Option ist dann minimal, wenn es keine andere Split Option gibt, die den Graphen anhand einer Teilmenge von Kanten der zu prüfenden Split Option teilt.

Der Funktion „findSplitOption“ übergeben wir am Anfang den Graphen für den wir die Split Optionen berechnen wollen, eine Liste mit den Kanten des Graphen, eine leere Liste mit bereits entfernten Kanten und eine leere Liste mit Split Optionen. Abbildung 5.18 zeigt für einen Beispielgraphen die Vorgehensweise: Wir entnehmen die erste Kante aus „edgeQueue“ (Kante 1) und entfernen diese Kante aus dem Graphen. Da der Graph noch zusammenhängend ist, führen wir dies iterativ fort bis es zu einer Teilung des Graphen kommt (Kante 2 und 3). An dieser Stelle prüfen wir, ob die Lösung (bis zu dieser Stelle) minimal ist und fügen die Split Option zum „resultSet“ hinzu. Bei der Prüfung, ob eine Lösung

minimal ist, entfernen wir vorherige nicht minimale Lösungen aus dem resultSet (dargestellt durch Pfeile). Anschließend gehen wir in der Rekursionstiefe eine Ebene zurück und versuchen nun anstatt Kante 3 die nächste Kante aus der Liste zu entfernen (Kante 4). Falls es keine weitere Kante in der Liste mehr gibt gehen wir in der Rekursionstiefe solange zurück bis die Liste wieder Kanten enthält (oder der Algorithmus terminiert). Diese Vorgehensweise entspricht einer „Tiefensuche“. Die durchgestrichenen Lösungen sind entweder nicht minimal oder trennen den Graphen nicht. Es verbleiben folglich sechs minimale Split Optionen.

```
function List<SplitOption> findSplitOptions(Graph graph, List<Edge> edgeQueue, List<Edge> removeSet,
                                                          List<SplitOption> resultSet){
    while (edgeQueue not empty){
      //Nächste Kante aus der Schlange entnehmen:
      currentEdge = edgeQueue.removeFirst();

      //Kopien anfertigen:
      List<Edge> removeSetCopy = removeSet.clone();
      List<Edge> edgeQueueCopy = edgeQueue.clone();
      Graph graphCopy = graph.clone();

      //Aktuelle Kante zur Liste entfernter Kanten hinzufügen und alle Kanten dieser Liste aus dem Graphen entfernen:
      removeSetCopy.add(currentEdge);
      graphCopy.removeEdges(removeSetCopy);

      //Prüfen, ob der Graph noch zusammenhängend ist:
      if (not graphCopy.isConnected){
          //Graph wurde geteilt --> Prüfen, ob Lösung minimal ist, nicht minimale Lösungen aus resultSet entfernen:
          if (removeSetCopy minimal){
              //Lösung zu Ergebnis hinzufügen:
              resultSet.add(removeSetCopy);
              resultSet.removeNotMinimalResults();
          }
      } else {
          //Graph hängt noch zusammen --> weitere Kanten entfernen:
          findSplitOptions(graph, edgeQueueCopy, removeSetCopy, resultSet);
      }
    return resultSet;
}
```

Abbildung 5.17: Pseudocode: Algorithmus zum Finden aller Split Optionen

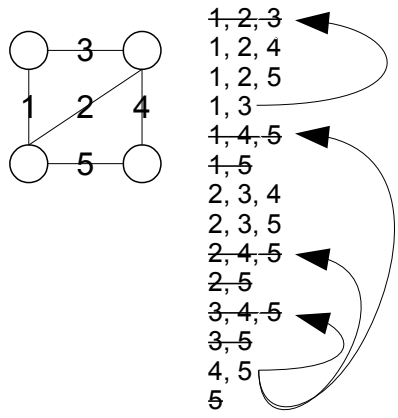


Abbildung 5.18: Vorgehensweise beim Finden sämtlicher Split Optionen

5.3 Gegenüberstellung der Ansätze

Sowohl der konstruktive als auch der korrektive Ansatz haben ihre Vor- und Nachteile. Während der konstruktive Ansatz aufgrund der Berechnung vieler Ähnlichkeitswerte und deutlich größeren Graphen recht aufwendig ist, hält sich der Aufwand beim korrektiven Ansatz in Grenzen, da hier die zu korrigierenden Komponenten deutlich kleiner sind und keine (oder nur wenige) Ähnlichkeitswerte berechnet werden müssen. Dafür kann der korrektive Ansatz Fehler jedoch nur innerhalb der durch die paarweisen Alignments vorgegebenen Komponenten korrigieren, sodass wir beim konstruktiven Ansatz insgesamt von einer höheren Qualität ausgehen können. Hinzu kommt, dass wir beim konstruktiven Ansatz nicht auf bereits existierende paarweise Alignments angewiesen sind.

6 Evaluation

In diesem Kapitel führen wir die Evaluation der im vorherigen Kapitel vorgestellten Ansätze durch. Das Ziel der Evaluation besteht darin zu zeigen, dass unsere Ansätze zur Berechnung multipler Alignments bessere Ergebnisse erreichen als gewöhnliche paarweise Alignments. Außerdem wollen wir die in Abschnitt 5.2 getroffene Vermutung evaluieren, dass der konstruktive Ansatz Ergebnisse höherer Qualität erzielt als der korrektive Ansatz. Die Evaluation führen wir anhand mehrerer Gold-Standards durch, die wir im ersten Abschnitt dieses Kapitels vorstellen. Im weiteren Verlauf evaluieren wir die verschiedenen Verfahren (konstruktiver und korrektiver Ansatz). Abschließend vergleichen wir die Ansätze bezüglich Qualität und Aufwand.

6.1 Gold-Standards

Ein wesentliches Ziel dieser Arbeit ist es, die Vorteile multipler Alignments gegenüber paarweisen Alignments herauszuarbeiten. Aus diesem Grund nutzen wir für die Evaluation der vorgestellten Ansätze bereits für die Evaluation von paarweisen Alignments verwendete Gold-Standards [Niemann and Gurevych, 2011; Meyer and Gurevych, 2011]. Dies ist zum einen ein aus 1809 Samples bestehender Gold-Standard zwischen WordNet und Wikipedia (englisch)[1] (kurz: WN-WP), zum anderen ein aus 2423 Samples bestehender Gold-Standard zwischen WordNet und Wiktionary (englisch)[2] (kurz: WN-WKT). Die Abbildungen 6.1 und 6.2 zeigen jeweils einen kurzen Ausschnitt aus den beiden Gold-Standards. Jede Zeile enthält ein WordNet - Wikipedia bzw. WordNet - Wiktionary Paar, das positiv (übereinstimmend=1) oder negativ (nicht übereinstimmend=0) annotiert ist. Die Annotation wurde anhand der zugehörigen Beschreibungstexte vorgenommen. Abbildung 6.3 zeigt die den letzten drei Zeilen von Abbildung 6.2 zugehörigen Beschreibungstexte, anhand derer eine Annotation vorgenommen wird.

```
# Each row represents a sense pair consisting of a WordNet 3.0 synset and a Wikipedia article
# annotated with "1" (same sense) or "0" (different sense)
# Synset offset  ; POS  ;  Synset  ; Wikipedia article title  ;  Annotation
5077146  ;  noun  ;  alignment  ;  Alignment (role-playing games)  ;  0
5077146  ;  noun  ;  alignment  ;  Alignment (Dungeons & Dragons)  ;  0
5077146  ;  noun  ;  alignment  ;  Alignment (political party)  ;  0
5077146  ;  noun  ;  alignment  ;  Alignment (archaeology)  ;  0
```

Abbildung 6.1: Ausschnitt aus dem Gold-Standard WordNet – Wikipedia

```
# WN synset offset ; pos ; lemma ; WKT id ; annotation
14036043 ; noun ; alertness ; 41114:0:1 ; 1
4664628 ; noun ; alertness ; 41114:0:1 ; 0
7445896 ; noun ; spread ; 7661:1:1 ; 1
7445896 ; noun ; spread ; 7661:1:2 ; 0
7445896 ; noun ; spread ; 7661:1:3 ; 0
```

Abbildung 6.2: Ausschnitt aus dem Gold-Standard WordNet – Wiktionary

[1] http://www.ukp.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/data/WordNetWikipedia1815Pairs_01.txt
[2] http://www.ukp.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/people/ChM/ijcnlp2011-meyer-dataset.txt

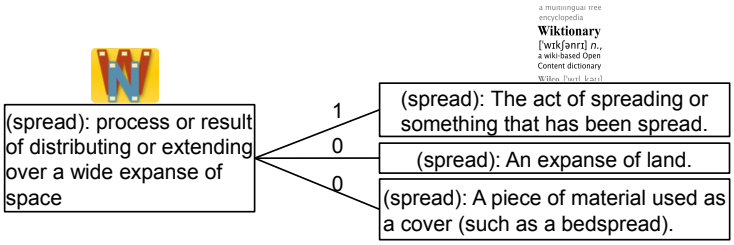


Abbildung 6.3: Annotation für einen paarweisen Gold-Standard

Anhand einer Analyse bezüglich dem Aufbau der beiden Gold-Standards ermitteln wir, dass zu 320 bzw. 484 WordNet Synsets im Durchschnitt 5,6 bzw. 5,0 Senses aus Wikipedia bzw. Wiktionary mit „1“ oder „0“ (übereinstimmend=positiv oder nicht übereinstimmend=negativ) annotiert sind. Die Anzahl der Zuweisungen pro Synset ist nicht beschränkt. Für WordNet-Wikipedia haben wir 227 positive Annotationen („1“) und 1582 negative Annotationen („0“), bei WordNet-Wiktionary sind es 313 positive bzw. 2110 negative Annotationen.

Zusätzlich zu den beiden gegebenen Gold-Standards haben wir zwei weitere Gold-Standards zwischen Wikipedia (englisch) und Wiktionary (englisch) (kurz: WP-WKT) automatisch aus diesen beiden Gold-Standards generiert. Dies ist möglich, weil die beiden Datensätze auf der gleichen Synset-/Lemma-Auswahl basieren. Dabei treffen wir die folgende Annahme: Wenn ein WordNet Synset A einem Wikipedia Sense B zugewiesen wird und das gleiche WordNet Synset A einem Wiktionary Sense C zugewiesen wird, dann sind auch Wikipedia Sense B und Wiktionary Sense C einander zuzuweisen. Wenn andererseits A und B und/oder A und C eine negative Annotation haben, so erhält auch B und C eine negative Annotation (siehe Abbildung 6.4, mit gestrichelter Kante). Diese Annahme können wir deshalb treffen, weil wir Synonymie als transitiv ansehen (siehe Abschnitt 2.1). Im Gegensatz zu den automatisch berechneten und mit Fehlern behafteten paarweisen Alignments können wir bei manuell annotierten Daten davon ausgehen, dass die Anzahl der Fehler hier deutlich geringer ist und daher keine Fehler von einem Ausmaß, wie wir sie in Kapitel 4 beobachtet haben, entstehen. Aufgrund unterschiedlicher Granularitäten ist jedoch nicht auszuschließen, dass einige Annotationen zumindest zu hinterfragen sind.

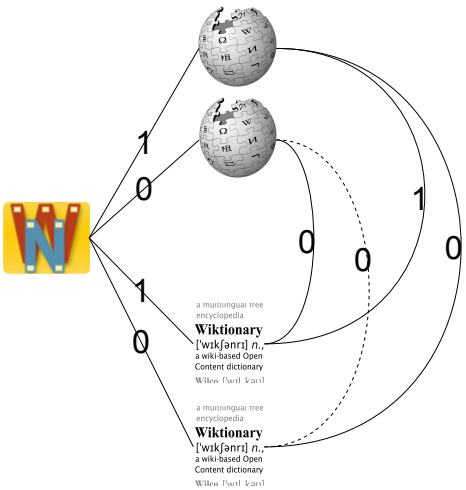


Abbildung 6.4: Erzeugung eines neuen Gold-Standards zwischen Wikipedia und Wiktionary (Kanten rechts) aus vorhandenen Gold-Standards (Kanten links). Die Zahlen „0“ und „1“ an den Kanten geben an, ob eine negative oder eine positive Annotation vorliegt

Dies führt zu einem neuen Gold-Standard mit 5586 Samples von denen 5499 eine negative und 87 eine positive Annotation darstellen. Wir kürzen diesen Gold-Standard im Folgenden mit „WP-WKT-A“ ab. Da der Anteil an positiven Annotationen hier sehr gering ist, haben wir noch einen weiteren Gold-Standard erstellt, indem wir als negative Annotationen nur diejenigen aufgenommen haben, für die entweder A und B oder A und C eine positive Annotation aufweisen (siehe Abbildung 6.4, ohne die gestrichtelte Kante). Der daraus resultierende Gold-Standard besteht aus 1084 Samples von denen 997 eine negative und (wie gehabt) 87 eine positive Annotation haben (Abkürzung: „WP-WKT-B). Abbildung 6.5 zeigt einen Ausschnitt aus dem neu generierten Gold-Standard zwischen Wikipedia (englisch) und Wiktionary.

```
# Wikipedia article title  ;  WKT id  ;  annotation
Abbey  ;  307:0:1  ;  1
Abbey  ;  307:0:2  ;  0
Abbey  ;  307:0:3  ;  0
Abbey (1922 automobile)  ;  307:0:1  ;  0
Abbey (bank)  ;  307:0:1  ;  0
```

Abbildung 6.5: Ausschnitt aus dem Gold-Standard Wikipedia – Wiktionary

Insgesamt stehen uns folglich vier Gold-Standards zur Evaluation paarweiser Alignments zur Verfügung. Tabelle 6.1 gibt eine Übersicht über die Gold-Standards. Zwar arbeiten die Gold-Standards für WordNet mit Synsets und für Wikipedia und Wiktionary mit Senses, dies entspricht jedoch unserer Pseudosynset Definition (siehe Abschnitt 4.4.1), da wir die UBY Senses für WordNet zu Synsets zusammengefasst, bei Wikipedia und Wiktionary jedoch darauf verzichtet haben. Um mit diesen Gold-Standards unser multiples Alignment evaluieren zu können, prüfen wir für alle Samples aus den Gold-Standards, ob die beiden zu vergleichenden Pseudosynsets jedes Samples sich im multiplen Alignment in ein und der selben Komponente befinden. Da in einem multiplen Alignment alle Pseudosynsets einer Komponente als synonym angesehen werden, sind dann folglich auch die beiden zu vergleichenden Pseudosynsets im multiplen Alignment einander zugewiesen. Auf diese Weise können wir einen direkten Vergleich der Qualität von paarweisen Alignments und multiplen Alignments durchführen.

	WN-WP	WN-WKT	WP-WKT-A	WP-WKT-B
# Samples	1809	2423	5586	1084
# 1-Annotationen	227	313	87	87
# 0-Annotationen	1582	2110	5499	997
Ø# Kandidaten	5,6	5,0	4,7	1,2

Tabelle 6.1: Übersicht über die vier verwendeten Gold-Standards

6.2 Konstruktiver Ansatz

Für die Evaluation des konstruktiven Ansatzes ist es zunächst notwendig eine Lösung für das in Abschnitt 5.1.2 beschriebene Problem der Komplexität des Ansatzes zu finden. Um den Rechenaufwand einigermaßen in Grenzen zu halten, ist eine Möglichkeit Kandidaten zu extrahieren unabdingbar. Da wir diesbezüglich über keine wirklich zufriedenstellende Methode verfügen und die Erforschung einer solchen nicht im Fokus dieser Arbeit steht, wählen wir die in Abbildung 6.6 illustrierte Vorgehensweise.

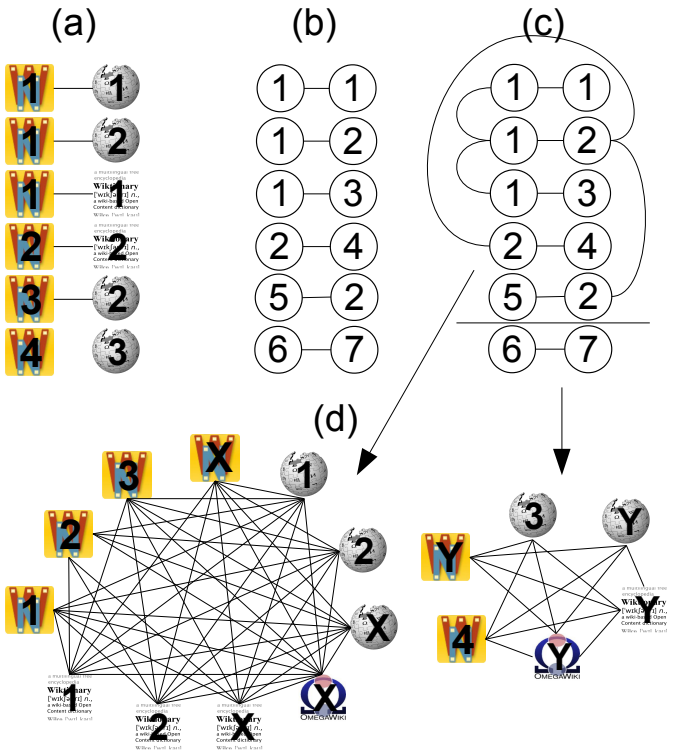


Abbildung 6.6: Konstruktion von Graphen aus dem Gold-Standard zur Evaluation des konstruktiven Ansatzes

Wir suchen uns zunächst für alle Synsets bzw. Senses aus den Gold-Standards die zugehörigen Pseudosynsets aus UBY heraus (a). In Abschnitt 4 haben wir die einzelnen über paarweise Alignments zusammenhängenden UBY Komponenten analysiert. Nun suchen wir uns diejenigen Komponenten heraus, welche die zuvor gefundenen Pseudosynsets enthalten (b). Diese Komponenten enthalten auch Pseudosynsets, die nicht im Gold-Standard enthalten sind (in der Abbildung mit X bzw. Y bezeichnet). Anschließend identifizieren wir die über den Gold-Standard miteinander verbundenen Komponenten (c) und extrahieren die darin enthaltenen Pseudosynsets, für die wir dann sämtliche paarweisen Ähnlichkeitswerte berechnen und daraus einen voll verbundenen, gewichteten Graphen erstellen (d). Ausgenommen sind Kanten zwischen Pseudosynsets der gleichen Ressource (deren Gewicht wird auf 0 gesetzt was in der Regel dazu führt, dass die Kanten entfernt werden). Aufgrund der Übersichtlichkeit haben wir in Abbildung 6.6 darauf verzichtet Kantengewichte einzuzeichnen.

Auf diese Weise erhalten wir für alle über den Gold-Standard zusammenhängende Komponenten je einen zusammenhängenden, gewichteten Graphen, der zu einem Pseudosynset im Gold-Standard sämtliche im Gold-Standard vorhandene Kandidaten enthält. Auf jedem dieser Graphen können wir dann die in Abschnitt 5.1.4 vorgestellten Clustering-Algorithmen oder den Topological Overlap (Abschnitt 5.1.5) anwenden, um unser multiples Alignment zu berechnen. In der Regel ist es jedoch

sinnvoll, den Graphen durch ein „Pruning“ zuvor etwas auszudünnen, indem besonders gering gewichtete Kanten (alle Kanten mit einem Gewicht unterhalb eines Schwellenwerts) vor Anwendung der Clustering-Algorithmen entfernt werden. Je höher der Schwellenwert ist, desto stärker wird das Pruning (desto mehr Kanten werden entfernt) und desto mehr gewinnt der Graph dadurch an Struktur - in dem Sinne, dass wir (im Gegensatz zu einem vollständig verbundenen Graphen) anhand dieser Struktur leichter erkennen können, welche Knoten ein Cluster bilden. Davon abgesehen verringert sich der Rechenaufwand für die Algorithmen, weil zahlreiche Kanten wegfallen.

6.2.1 Baseline

Um die in den folgenden Abschnitten vorgestellten Ergebnisse des konstruktiven Ansatzes einordnen zu können, benötigen wir zunächst noch eine Baseline. Wir vergleichen die Ergebnisse unseres multiplen Alignments mit denen eines einfachen paarweisen Alignments: Dazu berechnen wir uns für alle vier Gold-Standards die paarweisen Ähnlichkeiten aller darin angegebener Paare und trainieren jeweils einen Schwellenwert (bezüglich dem Maß „F-Measure“). Alle Paare mit einer Ähnlichkeit oberhalb des Schwellenwerts werden einander zugewiesen, alle anderen nicht. Auf Beschränkungen der Anzahl an Zuweisungen oder Bonussysteme, wie sie bei paarweisen Alignments oft genutzt werden, um die Ergebnisse zu verbessern, haben wir verzichtet, da dies die Vergleichbarkeit der Ergebnisse erschwert. Die Ergebnisse (siehe Tabelle 6.2) weichen aus diesem Grund von den in der Literatur zu findenden Werten [Niemann and Gurevych, 2011; Meyer and Gurevych, 2011] ab. Die auf dem größeren Gold-Standard zwischen Wikipedia (englisch) und Wiktionary (WP-WKT-A) ermittelten Ergebnisse müssen wir mit gewisser Vorsicht betrachten, weil der Anteil negativer Annotationen hier aufgrund der Vorgehensweise bei der automatischen Erzeugung des Gold-Standards erheblich größer ist als in den bereits existierenden Gold-Standards. Dadurch ist trotz sehr geringem F-Measure die Accuracy auf diesem Gold-Standard sehr hoch.

	WN-WP	WN-WKT	WP-WKT-A	WP-WKT-B
Schwellenwert (nicht normalisiert)	16,99	33,81	29,17	18,90
Schwellenwert (normalisiert)	64,16	66,53	73,98	66,16
Precision	38,31	60,00	19,44	35,89
Recall	70,04	61,34	24,14	54,02
Accuracy	82,09	89,72	97,26	88,56
F-Measure	49,53	60,66	21,54	43,12

Tabelle 6.2: Ergebnisse des paarweisen Alignments (alle Angaben in %)

Wir haben die Werte einmal mit und einmal ohne normalisierte Ähnlichkeitswerte berechnet (siehe Abschnitt 5.1.3). An den Ergebnissen ändert dies in diesem Fall nichts, weil die Normalisierungsfunktion streng monoton steigend ist. Wir erkennen in Tabelle 6.2 jedoch, dass sich die Schwellenwerte der unterschiedlichen Gold-Standards durch die Normalisierung einander annähern. Daraus schließen wir, dass die Normalisierung gut funktioniert, da ein Ähnlichkeitswert von etwa 0,65 sowohl zwischen Pseudosynsets aus WordNet und Wikipedia, als auch zwischen Pseudosynsets aus WordNet und Wiktionary bedeutet, dass das entsprechende Paar von Pseudosynsets eine Ähnlichkeit besitzt, die an der Grenze für eine Zuweisung liegt (d.h. die Ähnlichkeitswerte sind vergleichbar).

6.2.2 Hierarchisch Agglomeratives Clustering

Als Konfigurationsparameter stehen bei diesem Ansatz (siehe Abschnitt 5.1.4) das Pruning, sowie ein Schwellenwert als Abbruchbedingung zur Verfügung. In verschiedenen Experimenten haben wir festgestellt, dass es sinnvoll ist, vor jeder Iteration neu zu prunen (nicht nur einmalig vor der ersten Iteration). Dabei werden aus dem Graphen alle Kanten mit einem Ähnlichkeitswert kleiner als der Pruning-Wert entfernt. Sollte der Graph dadurch in mehrere Komponenten zerteilt werden, so wird der Algorithmus auf den einzelnen Komponenten fortgeführt, die dann in jedem Fall zwei unterschiedliche Cluster darstellen und in weiteren Iterationen nicht mehr zusammengeführt werden. Der Algorithmus terminiert, sobald der Graph keine Kante mehr enthält, deren Gewicht größer als der Schwellenwert ist. Die Ergebnisse verschiedener Konfigurationen sind Tabelle 6.3 zu entnehmen.

Schwellenwert	Pruning	WN-WP	WN-WKT	WP-WKT-A	WP-WKT-B
0,57	0,3	65,20	62,98	28,81	53,68
0,59	0,1	65,45	**63,81**	29,52	54,14
0,59	0,3	**65,76**	62,75	29,52	**54,44**
0,59	0,5	64,98	63,55	28,05	51,98
0,59	0,6	61,02	63,52	27,12	49,38
0,61	0,3	62,74	63,39	29,14	51,46
0,63	0,3	60,55	63,53	30,22	51,85
0,65	0,1	58,95	62,16	**30,35**	51,66
0,65	0,3	58,95	62,16	**30,35**	51,66
0,65	0,5	58,79	62,16	**30,35**	51,32
0,67	0,3	54,14	61,45	28,70	47,14
Baseline		49,53	60,66	21,54	43,12

Tabelle 6.3: Ergebnisse (F-Measure) mit verschiedenen Konfigurationen

Mit einem Schwellenwert zwischen 0,59 und 0,65 (leicht unterhalb der in den paarweisen Alignments trainierten Schwellenwerten) erreichen wir mit dem Hierarchisch Agglomerativen Verfahren die besten Ergebnisse (fett markiert in Tabelle 6.3). Bezüglich dem F-Measure Wert verzeichnen wir auf allen Gold-Standards eine Verbesserung zwischen 3% und 16% im Vergleich zur Baseline. Es wird zudem deutlich, dass die Höhe des Pruning-Werts bei diesem Verfahren keinen großen Einfluss auf die Ergebnisse hat, sofern er nicht zu hoch (>0.5) ist. Dies hängt damit zusammen, dass das Hierarchisch Agglomerative Verfahren als agglomeratives Verfahren immer zunächst die Kanten mit den höchsten Ähnlichkeitswerten betrachtet und Ähnlichkeitswerte zwischen Clustern in den einzelnen Iterationen automatisch reduziert werden, sodass sie unterhalb des Schwellenwertes liegen. Die Struktur des Graphen in dem Sinne, dass man anhand der Kanten bereits Cluster erkennen kann, ist (anders als beispielsweise beim Newman Clustering) weniger wichtig.

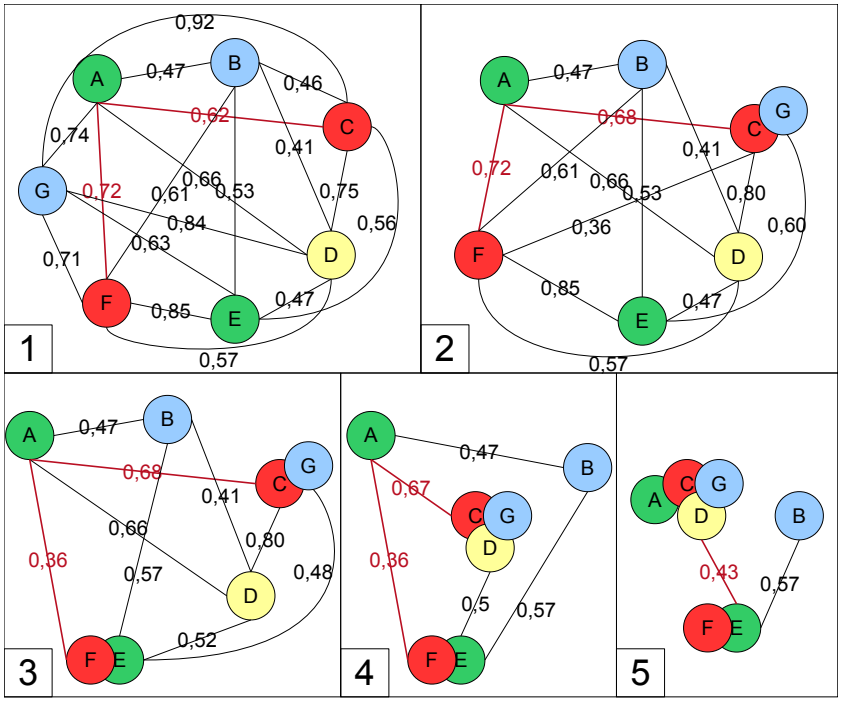
In dem Graphen auf dem wir arbeiten gibt es (wie schon beschrieben) keine Kanten zwischen Knoten der gleichen Ressource (da der Ähnlichkeitswert auf 0 gesetzt wird und entsprechende Kanten somit durch das Pruning entfernt werden). Wir haben dies mit dem Argument begründet, dass innerhalb eines Clusters nur in Ausnahmefällen mehrere Pseudosynsets der gleichen Ressource vorkommen sollten (siehe Abschnitt 4.4.2). Behandelt man hingegen alle Knoten im Graphen als gleich und berechnet auch zwischen Pseudosynsets der gleichen Ressource Ähnlichkeitswerte, so erhalten wir F-Measure Werte, die (bei allen Gold-Standards) etwa 10% unterhalb der Werte aus Tabelle 6.3 liegen. Die von uns gewählte Vorgehensweise erweist sich daher als sinnvoll.

Desweiteren haben wir uns dafür entschieden die UBY Senses zu Pseudosynsets zusammenzufassen, damit Wörter mit vielen Synonymwörtern den Graphen nicht „aufblähen“ (siehe Abschnitt 4.4.1). Führen wir das Verfahren hingegen direkt auf den UBY Senses durch, so reduzieren sich die erreichten F-Measure Werte wiederum um etwa 10%. Auch in diesem Punkt erweist sich folglich die gewählte Vorgehensweise Senses zunächst zu Pseudosynsets zusammenzufassen als begründet.

Diskussion und Fehleranalyse

Um die Wirkungsweise des Ansatzes beurteilen zu können und die häufigsten Fehlerklassen zu ermitteln, führen wir im Folgenden eine qualitative Auswertung des Verfahrens durch. In den folgenden Abbildungen stehen die unterschiedlichen Farben jeweils für eine bestimmte Ressource: grün=WordNet, rot=Wiktionary, blau=Wikipedia (englisch), gelb=Wikipedia(deutsch). Einander überlappende Knoten befinden sich in einem Cluster und werden einander zugewiesen.

Zunächst wollen wir anhand eines Positivbeispiels zeigen, wie die globale Struktur genutzt werden kann, um in einem paarweisen Alignment gemachte Fehler zu vermeiden. Abbildung 6.7 zeigt die einzelnen Iterationen des Hierarchisch Agglomerativen Clusterings.



3 Bedeutungen: Kaschubaum (A/C/D/G), Kaschunuss (E, F), U-Bahn Station (B)

Abbildung 6.7: Aufgrund der globalen Struktur werden A und C einander zugewiesen und nicht A und F (Schwellenwert: 0,65 / Pruning: 0,3)

Insgesamt haben wir drei unterschiedliche Bedeutungen in dem Ursprungsgraphen: Kaschubaum, Kaschunuss und eine U-Bahn Station. Interessant sind vor allem die rot markierten Kanten A/C und A/F: Die Kante A/C liegt mit einem Ähnlichkeitswert von 0,62 fälschlicherweise unterhalb des Schwellenwertes, während die Kante A/F mit einem Schwellenwert von 0,72 fälschlicherweise oberhalb des Schwellenwertes liegt. Dieser Fehler in den Ähnlichkeitswerten entsteht dadurch, dass in den Beschreibungen der Bedeutungen „Kaschubaum“ und „Kaschunuss“ relativ viele Wörter übereinstimmen. In einem einfachen paarweisen Alignment würden diese beiden Kanten daher falsch klassifiziert werden. Bei der Berechnung eines multiplen Alignments mit dem Hierarchisch Agglomerativen Verfahren hingegen erhöht sich das Kantengewicht für A/C durch den hohen Ähnlichkeitswert der Kante A/G in

der ersten Iteration. In der zweiten Iteration wird zudem das Gewicht von A/F reduziert, weil A und E aus der gleichen Ressource stammen und daher einen Ähnlichkeitswert von 0 haben. Der Fehler kann somit durch das Ausnutzen der globalen Struktur vermieden werden.

Trotz dieser positiven Effekte entstehen bei der Berechnung multipler Alignments mit dem Hierarchisch Agglomerativen Verfahren Fehler deren Ursachen wir anhand des Beispiels aus Abbildung 6.8 zeigen.

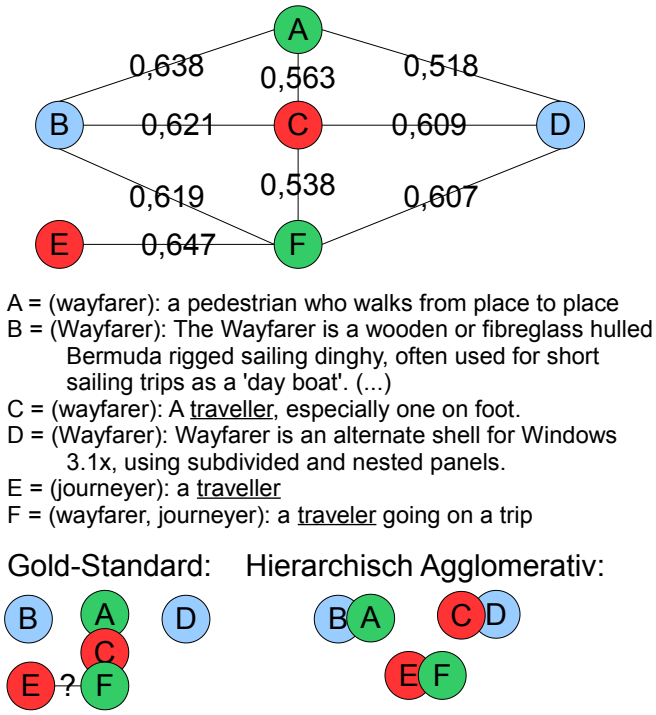


Abbildung 6.8: Die Ähnlichkeitswerte zwischen A/C/F sind zu niedrig, während die Ähnlichkeitswerte zu B bzw. D zu hoch sind (Schwellenwert: 0,6 / Pruning: 0,3)

Die wohl häufigste Fehlerursache sind Fehler in den berechneten paarweisen Ähnlichkeitswerten. In Abbildung 6.8 erkennen wir leicht, dass die beiden Wikipedia Pseudosynsets „Segelboot“ (Knoten B) bzw. „Windows-Shell“ (Knoten D) völlig andere Bedeutungen haben als die restlichen Pseudosynsets der Ressourcen WordNet und Wiktionary (Knoten A, C, E, F), die einen „Reisenden“ beschreiben. Dennoch beobachten wir eher hohe Ähnlichkeitswerte zwischen den Wikipedia Pseudosynsets und den anderen Pseudosynsets, während die Ähnlichkeitswerte innerhalb der WordNet und Wiktionary Pseudosynsets eher gering sind (mit Ausnahme von E/F). Unten links sehen wir die vom Gold-Standard als korrekt eingestuften Zuweisungen. Über das Pseudosynset zu Knoten E trifft der Gold-Standard keine Aussage, wir sehen es aufgrund der Beschreibungstexte jedoch den Knoten A, C und F zugeordnet. Unten rechts sehen wir die vom Hierarchisch Agglomerativen Verfahren (auf Basis der oben angegebenen Komponente mit den fehlerhaften Ähnlichkeitswerten) berechneten Zuweisungen.

Als Ursache für die fehlerhaften Ähnlichkeitswerte kommen mehrere Punkte in Betracht: Zum einen sind die Beschreibungstexte der Ressourcen WordNet und Wiktionary sehr kurz und beschreiben die zugehörige Bedeutung somit nicht genau genug. Da wir zudem ein Standardverfahren zur Berechnung der Ähnlichkeitswerte genutzt haben, das auf Wortüberlappung basiert (siehe Abschnitt 5.1.1), können wir zwei Bedeutungen nur dann als „ähnlich“ erkennen, wenn in deren Beschreibungstexten die gleichen Wörter verwendet werden. In diesem Beispiel finden wir in den sehr kurzen Beschreibungstexten

unterschiedliche Schreibweisen für das Wort „traveller“ (englisch) bzw. „traveler“ (amerikanisch), was geringere Ähnlichkeitswerte zur Folge hat. Eine mögliche Lösung um mit derartigen Problemen umzugehen, ist die Verwendung eines semantischen Ansatzes zur Berechnung von Ähnlichkeitswerten (z.B. Toral *et al.* [2009], Niemann and Gurevych [2011]). Das Problem zu kurzer und ungenauer Beschreibungstexte können wir damit jedoch nicht beheben. Zum Teil enthalten die Beschreibungstexte zudem Fehler, z.B. durch das Nutzen einer Bildunterschrift als Beschreibungstext, anstatt des ersten Absatzes eines Wikipedia Artikels. Hier ist die Datengrundlage zu verbessern.

Eine weiteres Problem bezüglich der Qualität von Ähnlichkeitswerten ist die Normalisierung der Ähnlichkeitswerte, die eine Vergleichbarkeit der Ähnlichkeitswerte zwischen unterschiedlichen Ressourcen-Paaren zum Ziel hat. In diesem Beispiel erkennen wir, dass die Vergleichbarkeit der Ähnlichkeitswerte nicht gegeben ist: Durch die Normalisierung erhalten wir recht hohe Ähnlichkeitswerte zwischen den Wikipedia Pseudosynsets und den übrigen Pseudosynsets, während die Ähnlichkeit zwischen den Pseudosynsets der übrigen Ressourcen eher gering ausfallen. Die Ressource Wikipedia nimmt aufgrund der langen Beschreibungstexte unter den Ressourcen eine Sonderrolle ein. So erhält man bei zwei Ressourcen mit recht kurzen Beschreibungstexten (z.B. WordNet und Wiktionary) entweder eine recht hohe oder eine recht niedrige Ähnlichkeit zweier Pseudosynsets, während es bei längeren Beschreibungstexten (z.B. WordNet und Wikipedia) öfter Abstufungen gibt. Tatsächlich ist die Standardabweichung der Ähnlichkeitswerte zwischen WordNet und Wiktionary etwa doppelt so hoch wie zwischen WordNet und Wikipedia. Möglicherweise würde es daher helfen, die Standardabweichung in die Normalisierung von Ähnlichkeitswerten einzubeziehen, um besser mit unterschiedlich langen Beschreibungstexten umgehen zu können.

Abgesehen von fehlerhaften oder nur schwer vergleichbaren Ähnlichkeitswerten gibt es weitere Fehlerursachen, die mit der Struktur der Komponenten auf denen wir arbeiten zusammenhängen. Teilweise besteht die Komponente, auf der wir das Hierarchisch Agglomerative Clustering ausführen, aus nur zwei Ressourcen. Hier können wir nicht von einer globalen Struktur profitieren und das multiple Alignment wird daher im Prinzip auf ein paarweises Alignment reduziert. Die Bestimmung von Kandidaten aus mehreren Ressourcen kann zu einer globalen Struktur beitragen. Ein anderes Problem wird durch Pseudosynsets verursacht, für die es kein synonymes Pseudosynset aus einer anderen Ressource gibt. In dem Beispiel aus Abbildung 6.8 trifft dies auf die beiden Wikipedia Pseudosynsets „Segelboot“ und „Windows-Shell“ zu. Gäbe es ein synonymes Pseudosynset einer anderen Ressource, so wäre vermutlich der Ähnlichkeitswert zu diesem Pseudosynset höher als zu den Pseudosynsets der Bedeutung „Reisender“, wodurch die Wikipedia Pseudosynsets zu dem synonymen Pseudosynset hingezogen würden und gleichzeitig weg von den nicht synonymen Bedeutungen. Weil Wikipedia mit über 3 Millionen Pseudosynsets weitaus mehr Bedeutungen abdeckt als alle anderen Ressourcen, tritt dieses Problem hier besonders häufig auf. Allerdings ist dies nur dann ein Problem, wenn die Ähnlichkeitswerte aufgrund einer unzureichenden Normalisierung nicht vergleichbar sind: Wären in Abbildung 6.8 die Ähnlichkeitswerte zwischen „Segelboot“ (Knoten B) und „Reisender“ (Knoten A, C, F) durch eine bessere Normalisierung und qualitativ hochwertigere Ähnlichkeitswerte niedriger, so würde das Pseudosynset „Segelboot“ vom Hierarschich Agglomerativen Verfahren isoliert werden.

Abschließend wollen wir noch einen Blick auf den Gold-Standard werfen. In Abschnitt 4.4.2 haben wir uns damit beschäftigt, wie mit unterschiedlichen Granularitäten der Ressourcen umzugehen ist und sind zu dem Schluss gekommen, dass in der Regel nicht mehrere Pseudosynsets der gleichen Ressource aligniert werden sollten. In dem Beispiel aus Abbildung 6.8 haben wir festgestellt, dass die Knoten A, C, E und F alle die Bedeutung des „Reisenden“ beschreiben. Andererseits könnte man die Pseudosynsets A und C („zu Fuß Reisender“) als etwas spezieller ansehen, als die Pseudosynsets E und F („Reisender“) und die beiden Bedeutungen gesondert betrachten. Dadurch könnte die Zuweisung mehrerer Pseudosynsets der gleichen Ressource vermieden werden. Der Gold-Standard sieht jedoch die

Pseudosynsets A, C und F als synonym an und trifft keine Aussage über Pseudosynset E. Wir stellen daher fest, dass es teilweise sehr deutlich ist, ob zwei Pseudosynsets synonym sind (z.B. „Segelboot“ vs. „Reisender“) und teilweise Aussagen streitbar sind („Reisender“, „zu Fuß Reisender“). Dementsprechend sind von den Verfahren gemachte Fehler prinzipiell auch als unterschiedlich gravierend zu bewerten.

Zusammenfassend können wir festhalten, dass die meisten Fehler durch Probleme mit den Ähnlichkeitswerten entstehen. Diese hängen mit der Datengrundlage (zu kurze Beschreibungstexte, Rauschen), dem Ansatz zur Berechnung von Ähnlichkeitswerten oder einer mangelnden Vergleichbarkeit von Ähnlichkeitswerten zwischen verschiedenen Ressourcen-Paaren (unzureichende Normalisierung) zusammen. Desweiteren ist die (globale) Struktur der Komponenten, auf denen das Hierarchisch Agglomerative Verfahren arbeitet, wichtig, was die Bedeutsamkeit einer geeigneten Kandidatenauswahl unterstreicht. Wir haben zudem festgestellt, dass Fehler unterschiedlich gravierend sein können. Es wäre beispielsweise möglich, dies bei der Evaluation multipler Alignments zu berücksichtigen.

6.2.3 Newman Clustering

Analog zum Hierarchisch Agglomerativen Verfahren stehen als Konfigurationsparameter Pruning-Wert und Schwellenwert zur Verfügung. Das Pruning wird einmalig vor der ersten Iteration durchgeführt, der Schwellenwert entspricht hier nicht den Ähnlichkeitswerten, sondern den für die Kanten berechneten Betweenness-Werten (welche in der gewichteten Version des Newman Clusterings durch das Kantengewicht geteilt werden). Je niedriger der Schwellenwert, desto stärker wird der Graph zerteilt und desto kleiner werden die Cluster. Außer einer Variation dieser beiden Konfigurationsparameter haben wir einige Experimente mit variablen Schwellenwerten durchgeführt. Dazu haben wir für jene Komponenten, die nach dem Clustering noch mehr als 6 Knoten enthalten, den Schwellenwert schrittweise um 0,2 gesenkt. Die Ergebnisse sind Tabelle 6.4 zu entnehmen.

Schwellenwert	Pruning	WN-WP	WN-WKT	WP-WKT-A	WP-WKT-B
2,0*	0,60	61,04	50,63	27,34	39,08
2,0	0,63	58,91	53,00	28,57	46,36
2,3*	0,50	**63,16**	41,04	**39,13**	51,72
2,3*	0,60	60,33	**55,05**	29,71	49,40
2,3*	0,63	60,61	55,01	29,66	49,06
2,5*	0,50	63,13	41,25	38,10	51,34
2,5*	0,55	61,11	43,90	35,42	51,34
2,5*	0,60	60,75	54,09	31,51	**51,98**
2,5	0,50	58,94	41,74	28,99	49,50
2,5	0,55	57,63	42,70	27,43	48,98
Baseline		49,53	60,66	21,54	43,12

Tabelle 6.4: Ergebnisse (F-Measure) mit verschiedenen Konfigurationen (* = Schwellenwert variabel)

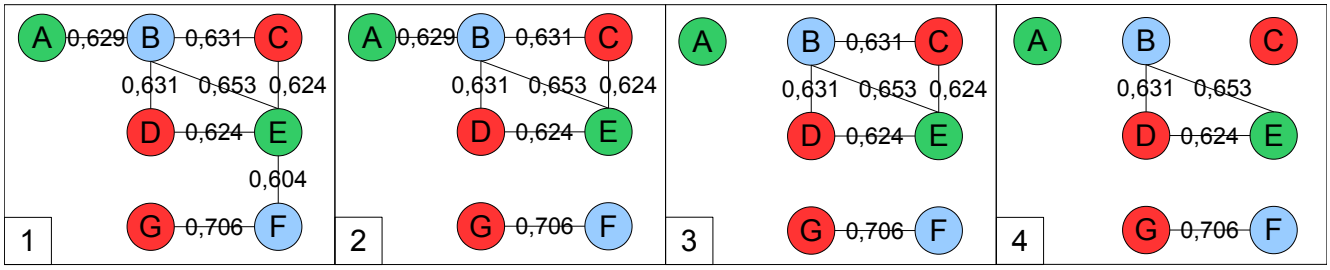
Wir erkennen, dass variable Schwellenwerte eine deutliche Verbesserung der Werte bewirken. Außerdem stellen wir fest, dass die Werte für den Gold-Standard WN-WKT deutlich schlechter als für die anderen Gold-Standards sind und sogar unterhalb der Baseline. So erhalten wir für die Gold-Standards WN-WP und WP-WKT zum Teil noch bessere Werte als beim Hierarchisch Agglomerativen Verfahren, gleichzeitig aber schlechte Werte für den Gold-Standard WN-WKT. Da wir in erster Linie an einem qualitativ hochwertigen multiplen Alignment interessiert sind und weniger daran paarweise Alignments zu optimieren, ist eine Konfiguration, die auf allen Gold-Standards vernünftige Werte liefert, daher vorzuziehen. Desweiteren wird bei diesem Ansatz ein stärkeres Pruning verwendet, das nur noch

geringfügig unterhalb des Schwellenwertes für paarweise Alignments liegt. Dies ist einerseits notwendig um die Komplexität des Algorithmus zu reduzieren, andererseits um eine Struktur in den Graphen zu bekommen, was bei diesem Algorithmus von großer Relevanz ist.

Wenn wir einen etwas genaueren Blick in die Daten werfen, so stellen wir fest, dass für die schlechten Werte auf dem Gold-Standard WN-WKT in erster Linie ein schlechter Recall verantwortlich ist bzw. eine zu hohe Anzahl an False Negatives. Wir werden in der Fehleranalyse darauf eingehen, wieso ein schwächeres Pruning scheinbar die Anzahl der False Negatives erhöht.

Fehleranalyse

Abbildung 6.9 zeigt die einzelnen Iterationen des Newman Clusterings für eine gegebene Komponente. Wir werden anhand dieses Beispiels im Folgenden einige typische Eigenschaften und Probleme des Newman Clusterings vorstellen. Die Komponente besteht aus 6 verschiedenen Bedeutungen, wobei lediglich die durch die Knoten F und G repräsentierten Pseudosynsets synonym und somit einander zuzuweisen sind.



A = (stringer): a member of a squad on a team
B = (Longeron): Interior of a Boeing/Stearman PT-17 showing small channel section stringers.
C = (stringer): Someone who leads someone along.
D = (stringer): Someone who threads something.
E = (stringer): a worker who strings
F = (Stringer): In journalism, a stringer is a type of freelance journalist who contributes reports to a news organization on an on-going basis but is paid individually for each piece of published or broadcast work.
G = (stringer): A freelance correspondent not on the regular newspaper staff, especially one retained on a part-time basis to report on events in a particular place.

Abbildung 6.9: Einzelne Iterationen des Newman Clusterings (Schwellenwert: 2,3 (variabel) / Pruning: 0,6)

Der Graph aus Abbildung 6.9 erhält durch das Pruning eine Struktur, die prinzipiell sehr gut für den Newman Algorithmus geeignet ist. So werden zunächst die Kanten E/F und A/B entfernt. Wir erhalten für alle Kanten von voll vernetzten Graphen immer den Betweenness-Wert 1. Da wir diesen anschließend durch das jeweilige Kantengewicht teilen, wird eine Kante in einem voll vernetzten Graphen nur bei einem Gewicht kleiner als 0,43 entfernt (bei Schwellenwert 2,3), was niemals vorkommt, da eine solche Kante in der Regel schon vorher durch Pruning entfernt wurde. Werden also zwei Knoten (z.B. G und F) vom Rest des Graphen abgeschnitten, so bleiben diese beiden Knoten in jedem Fall verbunden. In der vierten Iteration wird Knoten C von den Knoten B, D, E abgetrennt. Danach sind die Knoten B, D, E voll vernetzt, sodass hier nicht weiter geteilt wird. Allerdings sind die Ähnlichkeitswerte für B/C und B/D, sowie für E/C und E/D bis auf drei Nachkommastellen gleich, sodass man in dieser Iteration auch den Knoten D hätte abtrennen können (anstatt C). Generell ist festzustellen, dass mit zunehmender Anzahl an entfernten Kanten die verbleibenden zusammenhängenden Komponenten immer stärker vernetzt sind, wodurch das Entfernen weiterer Kanten unwahrscheinlicher wird.

Das Newman Verfahren tendiert zudem dazu, Cluster zu bilden, die möglichst aus jeder Ressource genau einen Knoten enthalten, da es keine voll vernetzten Teilgraphen, bestehend aus mehreren Ressourcen, geben kann (keine Kanten zwischen Pseudosynsets der gleichen Ressource). In diesem Beispiel führt diese Eigenschaft des Newman Clusterings dazu, dass die Knoten B, D und E einander zugewiesen werden. Ein Blick auf die Beschreibungstexte zeigt, dass es sich hierbei um drei unterschiedliche Bedeutungen handelt (B=Begriff aus Verkehrstechnik, D=Auffädeln, E=Arbeiter, der etwas aufspannt). Die Ähnlichkeitswerte dieser drei Kanten sind dennoch relativ hoch und der Graph zu stark vernetzt, um hier weiter aufzuteilen. Insbesondere der Ähnlichkeitswert zwischen B und E ist zu hoch, obwohl in den Beschreibungstexten lediglich ein Wort übereinstimmt, was erneut ein Hinweis auf Probleme bei der Normalisierung sein könnte. Ein semantischer Ansatz zur Berechnung paarweiser Ähnlichkeitswerte wäre hier in der Lage nicht nur auf die übereinstimmenden Wörter zu achten, sondern auch zu berücksichtigen in wie weit sich die restlichen Wörter unterscheiden.

Das Newman Verfahren nutzt sehr stark die durch das Pruning entstehende Topologie, weshalb der Ansatz auch ein stärkeres Pruning benötigt als beispielsweise das Hierarchisch Agglomerative Verfahren. Wir haben in den Experimenten festgestellt, dass sich bei einem etwas schwächeren Pruning die F-Measure Werte für den Gold-Standard WN-WKT stark verschlechtern, was mit einer hohen Anzahl an False Negatives zusammenhängt. Um zu erklären, wie die Stärke des Prunings mit der Anzahl an False Negatives zusammenhängt, betrachten wir das Beispiel aus Abbildung 6.10.

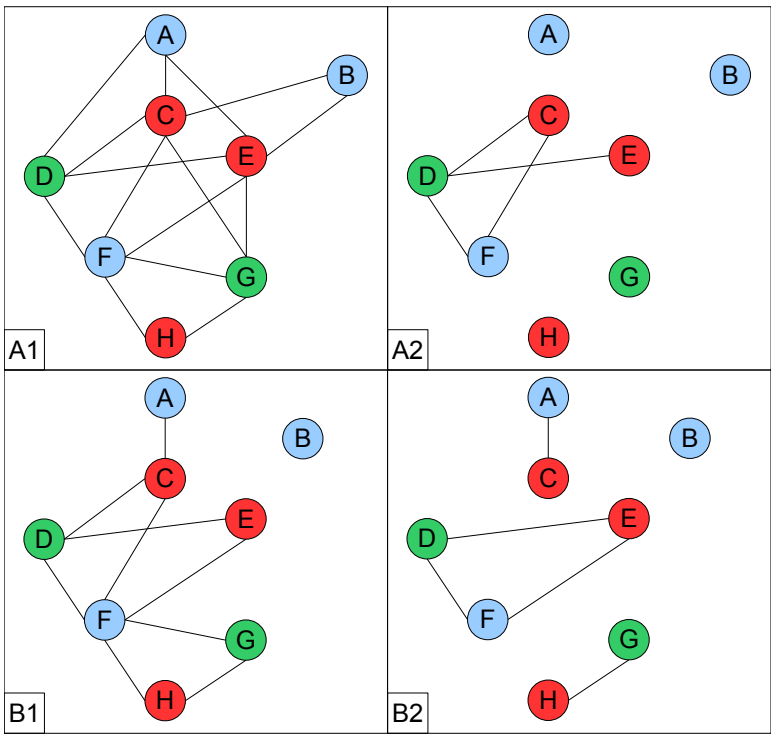


Abbildung 6.10: Oben (A): Pruning=0,5 / unten (B): Pruning=0,6 / links (1): Ausgangsgraph / rechts (2): Ergebnisgraph. Bei schwächerem Pruning ergeben sich mehr isolierte Komponenten.

Wir erkennen, dass bei schwächerem Pruning (oben) deutlich mehr isolierte Knoten entstehen, als bei etwas stärkerem Pruning (unten). Tatsächlich beträgt der Anteil isolierter Knoten (bei gleichem Threshold von 2,3) 87,32% bei Pruning mit 0,5 und nur 70,03% bei Pruning mit 0,6. Die Anzahl der Komponenten steigt bei weniger starkem Pruning um etwa 20% an. Die Größe der Komponenten ist bei stärkererm Pruning im Durchschnitt entsprechend um etwa 20% erhöht, wobei Komponenten mit Wiktionary Knoten stärker davon betroffen sind als andere Komponenten. Wie schon in Abschnitt 6.2.2

beschrieben, hängt dies mit der Standardabweichung zusammen, die für Ähnlichkeitswerte zwischen WordNet und Wiktionary etwa doppelt so groß ist wie zwischen WordNet und Wikipedia.

6.2.4 Topological Overlap

Neben den Clustering-Algorithmen haben wir das Maß „Topological Overlap“ vorgestellt (siehe Abschnitt 5.1.5), das in der Lage ist, Ähnlichkeitswerte in einem Graphen entsprechend dessen Topologie nach oben oder nach unten zu korrigieren. Wir können dieses Verfahren folglich nutzen, um die berechneten paarweisen Ähnlichkeitswerte mit Hilfe der globalen Struktur zu korrigieren, bevor wir auf dem korrigierten Graphen die vorgestellten Clustering-Algorithmen anwenden. Die Ergebnisse sind Tabelle 6.5 zu entnehmen.

	WN-WP	WN-WKT	WP-WKT-A	WP-WKT-B
Hierarchisch Aggl. (0,59 / 0,3)	62,43 (65,76)	25,82 (62,75)	45,01 (29,52)	47,41 (54,44)
Hierarchisch Aggl. (0,61 / 0,3)	62,50 (62,74)	20,00 (63,39)	47,41 (29,14)	49,61 (51,46)
Newman Clustering (2,3* / 0,6)	64,00 (60,33)	55,53 (55,05)	38,14 (29,71)	58,06 (49,40)

Tabelle 6.5: Ergebnisse (F-Measure) bei Korrektur von Ähnlichkeitswerten mit Topological Overlap (die Werte in Klammern entsprechen in der linken Spalte dem gewählten Schwellenwert bzw. dem Pruning-Wert, in den rechten Spalten dem Ergebnis der entsprechenden Konfiguration des Hierarschisch Agglomerativen bzw. Newman Verfahrens ohne Topological Overlap. * = variabler Schwellenwert).

Während sich das Maß Topological Overlap in Kombination mit dem Hierarchisch Agglomerativen Verfahren negativ auswirkt, beobachten wir für das Newman Clustering eine Verbesserung der Werte. Dies hängt damit zusammen, dass sowohl das Newman Clustering als auch das Maß Topological Overlap sehr stark auf der Topologie des Graphen aufbauen und daher ein starkes Pruning benötigen. Insofern ergänzen die beiden Verfahren sich hier sehr gut. Das Maß Topological Overlap arbeitet auf dem recht stark geprunten Graphen für das Newman Clustering deutlich besser als auf dem schwächer geprunten Graphen für das Hierarchisch Agglomerative Verfahren. In letzterem Fall führt das Maß aufgrund der zu gering ausgeprägten Topologie eher zu einem Angleichen der Ähnlichkeitswerte als zu einer wirklichen Korrektur.

6.3 Korrektiver Ansatz

Wie in Abschnitt 5.1 erläutert, basiert der korrektive Ansatz auf den über paarweise Alignments gebildeten Komponenten. Ausschließlich innerhalb dieser Komponenten kann eine Fehlerkorrektur durchgeführt werden. Im Folgenden beschreiben wir zunächst die Baseline für diesen Ansatz und besprechen anschließend die daraus resultierenden Ergebnisse.

6.3.1 Baseline

Ziel des korrektiven Ansatzes ist es durch paarweise Alignments gebildete Komponenten durch das Hinzufügen und Entfernen von Kanten zu korrigieren. Die Baseline sind somit die gegen die Gold-Standards evaluierten, in UBY enthaltenen paarweisen Alignments (siehe Tabelle 6.6). Wir haben hier daher andere Werte als bei der Evaluation des konstruktiven Ansatzes, wo wir selbst ein paarweises Alignment berechnet haben (siehe Abschnitt 6.2.1).

Da wir mit diesem Ansatz nur Fehler innerhalb der gegebenen Komponenten korrigieren können und auch dort nur dann, wenn wir diese Komponenten als fehlerhaft identifiziert haben, betrachten wir die Samples, für die eine Fehlerkorrektur mit diesem Verfahren überhaupt möglich ist, nochmals gesondert (jeweils der rechte Wert zu den Gold-Standards in Tabelle 6.6). Ein Pseudosynset Paar aus dem Gold-Standard lässt sich mit diesem Ansatz nur dann korrigieren, wenn sich die beiden Pseudosynsets vor der Korrektur in der gleichen Komponente befinden und wir mindestens einen Fehler in dieser Komponente identifiziert haben. Wir beschränken uns zudem auf die Gold-Standards WN-WP und WN-WKT, weil UBY kein paarweises Alignment zwischen Wikipedia und Wiktionary enthält, sodass wir auch keine Baseline vorliegen haben.

	WN-WP		WN-WKT	
	gesamt	fehlerhaft	gesamt	fehlerhaft
True Positives	101	24	175	45
True Negatives	1547	5	2036	45
False Positives	35	21	74	34
False Negatives	126	1	138	6
Precision	74,26	53,33	70,28	56,96
Recall	44,49	96,00	55,91	88,24
Accuracy	91,10	56,86	91,25	69,23
F-Measure	55,65	68,57	62,28	69,23

Tabelle 6.6: In UBY enthaltene paarweise Alignments (fehlerhaft: nur als fehlerhaft identifizierte Komponenten)

Auch bei dieser Baseline weichen die Werte aufgrund einer anderen Datengrundlage von den in der Literatur zu findenden Werten [Niemann and Gurevych, 2011; Meyer and Gurevych, 2011] ab. Da wir an dieser Stelle jedoch in erster Linie zeigen wollen, dass der korrektive Ansatz Fehler korrigieren und somit die Daten auf denen er aufbaut verbessern kann, stellt dies kein großes Problem dar.

6.3.2 Ergebnisse

Zunächst benötigen wir einen Algorithmus, der Möglichkeiten findet einen Graphen in zwei Komponenten zu teilen. Für Graphen mit >20 Kanten verwenden wir dazu den Newman Algorithmus für ungewichtete Graphen (siehe Abschnitt 5.1.4). Dieser findet jedoch in der Regel nur eine einzige Option. Für Graphen mit ≤20 Kanten ermitteln wir mit einem eigens entwickelten Algorithmus (siehe Abschnitt 5.2.1) sämtliche Split Optionen. Da der Algorithmus sehr aufwendig ist, können wir ihn nur für kleine Graphen nutzen. Aus den verschiedenen Optionen (Kanten entfernen, Graph teilen vs. Kanten hinzufügen, Graph nicht teilen) wählen wir wie in Abschnitt 5.2 beschrieben die Option mit der geringsten Anzahl an Edit-Operationen. Für den Fall, dass mehrere Optionen die gleiche Anzahl an Edit-Operationen durchführen, gibt es die Möglichkeit die Option mit der geringsten „Summe entfernter Ähnlichkeitswerte“ (SeÄ) auszuwählen. Dadurch werden jedoch Optionen bevorzugt, bei denen keine oder wenige Kanten entfernt werden: Wenn genau eine Kante fehlt, wird der Graph nie geteilt, weil es keine Option mit einer geringeren Anzahl an Edit-Operationen geben kann und die Summe entfernter Ähnlichkeitswerte 0 beträgt. Daher haben wir den Wert der „Summe entfernter Ähnlichkeitswerte“ für Optionen bei denen keine Kanten entfernt werden auf einen bestimmten Wert zwischen 0 und ∞ gesetzt. Setzen wir die „Summe entfernter Ähnlichkeitswerte“ auf ∞, so wird der Graph bei mehreren Optionen mit der gleichen Anzahl an Edit-Operationen immer geteilt. Die Ergebnisse der Experimente sind Tabelle 6.7 zu entnehmen. Wir erkennen für den Gold-Standard WN-WP eine deutliche Verbesserung im

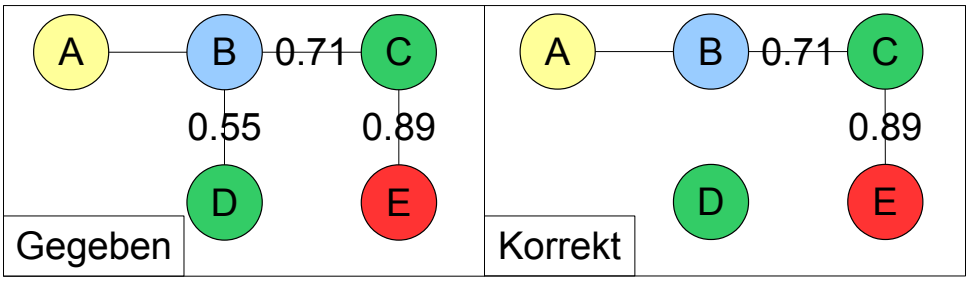
	WN-WP		WN-WKT		WP-WKT-A		WP-WKT-B	
SeÄ	gesamt	fehlerhaft	gesamt	fehlerhaft	gesamt	fehlerhaft	gesamt	fehlerhaft
0	56,11	71,64	62,46	69,57	37,35	41,86	46,97	62,07
0,5	55,87	70,77	62,57	70.07	37,58	43,90	46,97	62,07
0,55	56,34	74,19	62,54	70,15	37,04	42,11	46,15	59,26
0,6	56,09	73,33	62,52	**70,23**	37,50	44,44	46,51	61,54
0,65	56,09	73,33	62,25	69,29	38,75	50,00	47,69	66,67
0,7	56,25	**74,58**	62,23	69,35	38,46	50,00	46,88	64,00
0,75	56,25	**74,58**	62,23	69,35	39,74	**56,25**	48,06	**69,23**
0,8	55,01	67,86	62,34	69,92	37,66	46,67	45,67	58,33
∞	55,01	67,86	62,09	68,85	38,71	51,61	46,88	64,00
Baseline	55,65	68,57	62,28	69,23	-	-	-	-

Tabelle 6.7: Ergebnisse des korrektiven Ansatzes (SeÄ = Für die Summe entfernter Ähnlichkeitswerte gesetzter Wert). Zu jedem Gold-Standard sind zwei Werte (F-Measure) angegeben: Der erste Wert bezieht sich auf den gesamten Gold-Standard, der zweite Wert nur auf die Samples, bei denen der korrektive Ansatz prinzipiell in der Lage ist eine Korrektur durchzuführen.

Vergleich zur Baseline während wir bei WN-WKT hingegen nur eine leichte Verbesserung von etwa 1% erreichen. Auch für die beiden anderen Gold-Standards erreichen wir vernünftige Werte, die wir ohne Baseline jedoch nur schwer einordnen können. Bei den Daten ist zu berücksichtigen, dass der korrektive Ansatz sich nur auf einen sehr geringen Teil der Samples aus dem Gold-Standard anwenden lässt (zwischen 2,5% und 5,4% der Samples). Daher müssen die Ergebnisse mit gewisser Vorsicht betrachtet werden. Außerdem hat die Korrektur dadurch auf dem gesamten Datensatz im Vergleich zur Baseline praktisch keine Auswirkung, da ein Großteil der Kanten unverändert bleibt. Wir erkennen jedoch, dass der Ansatz prinzipiell funktioniert und zu einer Verbesserung der Alignments beitragen kann. Eine höhere Anzahl an gegebenen paarweisen Alignments (paarweise Alignments zwischen allen Paaren an Ressourcen) könnte ermöglichen, den Ansatz auf einen größeren Teil der Daten anzuwenden, da dann (durch die stärkere Vernetzung) vermutlich weniger Senses in verschiedenen Komponenten vorkommen, obwohl sie einander zuzuweisen sind.

Ähnlich wie beim konstruktiven Ansatz führen wir auch hier eine qualitative Auswertung des Ansatzes durch, um die Wirkungsweise beurteilen zu können und häufige Fehlerklassen zu identifizieren. Wie gehabt stehen in den folgenden Abbildungen die unterschiedlichen Farben jeweils für eine bestimmte Ressource: grün=WordNet, rot=Wiktionary, blau=Wikipedia (englisch), gelb=Wikipedia(deutsch).

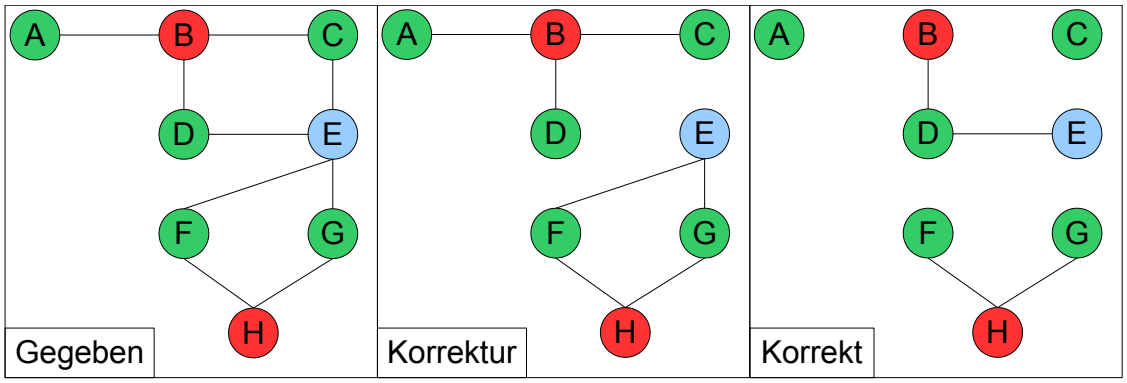
Zunächst betrachten wir ein Positivbeispiel, bei dem eine von dem Fehlerindikator als fehlerhaft eingestufte Komponente durch das Verfahren korrigiert wird. Die Komponente aus Abbildung 6.11 besteht aus den beiden unterschiedlichen Bedeutungen „Vormagen bei Wiederkäuern“ (Knoten A, B, C, E) und „Bauch“ (Knoten D). Knoten D ist somit von den übrigen Knoten der Komponente abzutrennen. Das korrektive Verfahren identifiziert in der gegbenen Komponente eine fehlende Kante (D/E). Neben dem Hinzufügen der Kante D/E (No-Split Option, Summe entfernter Ähnlichkeitswerte = 0,75) haben wir die Möglichkeit den Graphen durch das Entfernen einer Kante zu korrigieren (B/D oder B/C oder C/E). Die Split Option die Kante B/D zu entfernen hat mit 0,55 die geringste Summe entfernter Ähnlichkeitswerte unter den Split Optionen, sodass wir den Graphen korrekterweise an dieser Stelle in zwei Komponenten teilen, in denen wir keine fehlerhaften Kanten mehr identifizieren.



A = (Pansen): Der Pansen (lat. , über frz. „Wanst“; anatomisch Rumen) ist ein Hohlorgan bei Wiederkäuern (Ruminantia) und der größte der drei Vormägen. Er ist eine große Gärkammer, welche dem eigentlichen Drüsenmagen (bei Wiederkäuern als Labmagen bezeichnet) vorgeschaltet ist. (...)
B = (Rumen): The rumen, also known as a paunch, forms the larger part of the reticulorumen, which is the first chamber in the alimentary canal of ruminant animals. It serves as the primary site for microbial fermentation of ingested feed. The smaller part of the reticulorumen is the reticulum, which is fully continuous with the rumen, but differs from it with regard to the texture of its lining.
C = (first stomach): the first compartment of the stomach of a ruminant
D = (belly): a protruding abdomen
E = (rumen): The first stomach of ruminants; the paunch; the fardingbag.

Abbildung 6.11: Erfolgreiche Korrektur einer fehlerhaften Komponente

Die häufigsten Fehlerklassen verdeutlichen wir im Folgenden anhand des Beispiels aus Abbildung 6.12. Alle 8 Pseudosynsets der Komponente aus Abbildung 6.12 beschreiben das Wort „shank“, allerdings in vier unterschiedlichen Bedeutungen: „Oberes Ende eines Bohrers“ (Knoten A), „Schaft“ (Knoten C), „Schenkel (Fleisch)“ (Knoten B, D, E) und „Schenkel (Anatomie)“ (Knoten F, G, H). Ziel des korrektiven Ansatzes ist es, diese Komponente in ihre einzelnen Bedeutungen zu zerlegen.



A = (shank): cylinder forming the part of a bit by which it is held in the drill
B = (shank): Meat from that part of an animal.
C = (shank): cylinder forming the part of a bolt between the thread and the head
D = (shank): a cut of meat (beef or veal or mutton or lamb) from the upper part of the leg
E = (Shank): A meat shank or shin is the portion of meat around the tibia of the animal, the leg bone beneath the knee. (...)
F = (shank): lower part of the leg extending from the hock to the fetlock in hoofed mammals
G = (shank): the part of the human leg between the knee and the ankle
H = (shank): The lower part of the leg; shin.

Abbildung 6.12: Fehlerhafte Korrektur: Mehrere WordNet Pseudosynsets unterschiedlicher Bedeutung werden fälschlicherweise mit einem Wiktionary Pseudosynset aligniert

Der Fehlerindikator nimmt eine zentrale Rolle beim korrektiven Ansatz ein. Wir haben jedoch bereits in Abschnitt 4.1 festgestellt, dass der von uns verwendete Fehlerindikator nicht alle Fehlerarten identifizieren kann. So ist es beispielsweise nicht möglich Fehler in Komponenten mit nur zwei Ressourcen festzustellen. Dies führt zu der in Abbildung 6.12 gezeigten Korrektur: Die aus den Knoten A, B, C und D bestehende Komponente besteht aus nur zwei Ressourcen, sodass die Komponente als korrekt angesehen wird und von den anderen Knoten durch das Entfernen von zwei Kanten (zwischen E/D und E/C) abgetrennt wird. Betrachten wir die zugehörigen Bedeutungen „Oberes Ende eines Bohrers“, „Schaft“ und „Schenkel (Fleisch)“, so erkennen wir jedoch leicht, dass hier Fehler vorliegen. Eine mögliche Lösung für das Problem könnte darin bestehen, den Fehlerindikator dahingehend zu erweitern, dass Komponenten, in denen mehrere Pseudosynsets der gleichen Ressource zusammengefasst werden, grundsätzlich als fehlerhaft gelten. Allerdings ist auch diese Lösung nicht immer korrekt, da eine solche Komponente in manchen Fällen (aufgrund unterschiedlicher Granularitäten, siehe Abschnitt 4.4.2) auch korrekt sein kann: Die Knoten F, G und H beschreiben alle die Bedeutung „Schenkel (Anatomie)“, allerdings unterschiedlich spezifisch: Während Wiktionary die Bedeutung allgemein beschreibt (Knoten H), unterscheidet WordNet zwischen einem menschlichen Schenkel (Knoten G) und dem Schenkel eines Tieres (Knoten F).

Abgesehen von Schwächen der Fehlerindikatoren ist der korrektive Ansatz nur in der Lage mit einer begrenzten Anzahl an Fehlern umzugehen. In der in Abbildung 6.12 gegebenen Komponente sind insgesamt fünf fehlerhafte Kanten (vergleiche „Gegeben“ und „Korrekt“) von insgesamt neun Kanten. Eine geringere Anzahl an Fehlern in den Daten, auf denen der Ansatz arbeitet, würde die Korrektur vereinfachen.

6.4 Zusammenfassung der Ergebnisse

In Abbildung 6.13 sehen wir die jeweils besten Ergebnisse der verschiedenen Verfahren für den konstruktiven Ansatz. Wir erkennen, dass das Hierarchisch Agglomerative Verfahren (ohne Topological Overlap Measure) auf allen Gold-Standards bessere F-Measure Werte erzielt als das paarweise Alignment. Das Newman Verfahren weist bei dem Gold-Standard WN-WKT Schwächen auf, liefert ansonsten, insbesondere mit Topological Overlap Measure, jedoch ebenfalls bessere Werte als das paarweise Alignment. Während sich das Topological Overlap Measure beim Hierarchisch Agglomerativen Verfahren eher negativ auswirkt, erreichen wir durch das Maß in Kombination mit dem Newman Verfahren eine Verbesserung.

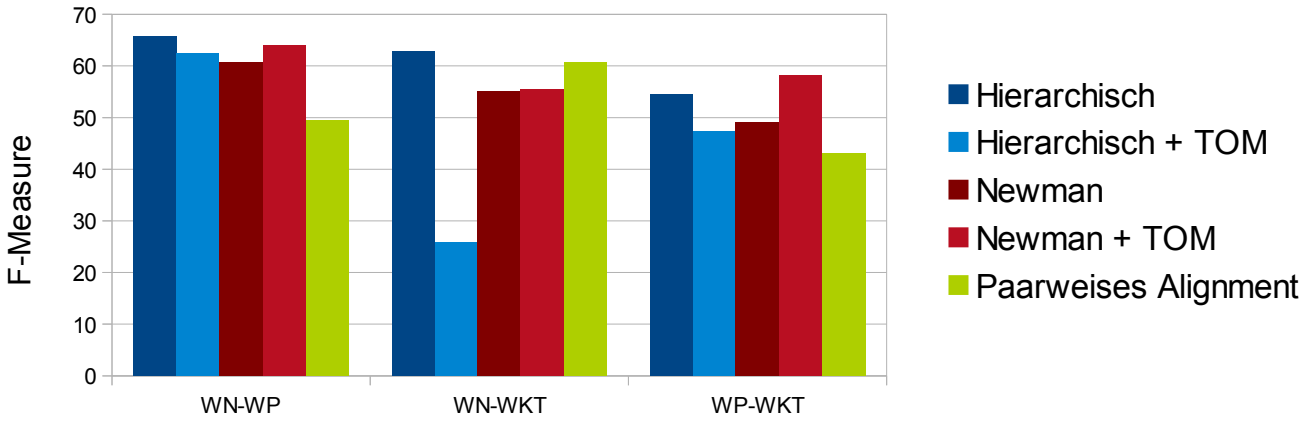


Abbildung 6.13: Ergebnisübersicht konstruktiver Ansatz (TOM = Topological Overlap Measure)

In Abbildung 6.14 sehen wir die Ergebnisübersicht für den korrektiven Ansatz. Auch hier erkennen wir in der besten Konfiguration eine Verbesserung der Werte im Vergleich zum paarweisen Alignment. Es ist jedoch zu beachten, dass die in der Evaluation ermittelten Werte auf einer recht geringen Anzahl an Komponenten basieren.

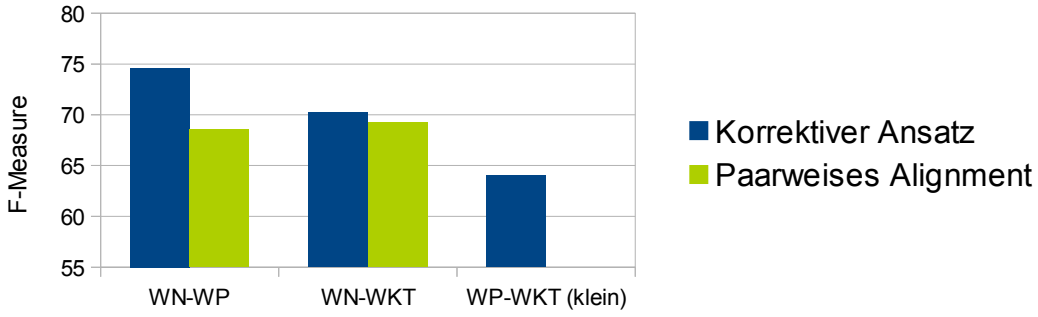


Abbildung 6.14: Ergebnisübersicht korrektiver Ansatz

7 Zusammenfassung

In dieser Masterarbeit wurden die Grundlagen multipler Alignments, also die Zuweisung von mehr als zwei lexikalisch-semantischen Ressourcen, erforscht. Das Ziel der Arbeit war es zu ergründen, ob ein multiples Alignment durch das Ausnutzen der durch mehr als zwei Ressourcen entstehenden globalen Struktur eine höhere Qualität erreicht als ein einfaches paarweises Alignment, bei dem lediglich zwei Ressourcen einander zugewiesen werden.

Wir haben zwei Ansätze zur Berechnung eines multiplen Alignments vorgestellt. Der konstruktive Ansatz berechnet ausgehend von einer Menge an Pseudosynsets unterschiedlicher Ressourcen Ähnlichkeitswerte zwischen diesen und baut daraus einen ungerichteten, gewichteten Graphen auf. Mit Hilfe von Clustering-Algorithmen (Hierarchisch Agglomerativ und Newman Clustering) oder Verfahren, welche die Ähnlichkeitswerte entsprechend der Topologie korrigieren (Topological Overlap), können dann die übereinstimmenden Pseudosynsets identifiziert werden. Dabei haben wir eine Reihe von Herausforderungen festgestellt. Zunächst ist der Aufwand für die Berechnung derart vieler Ähnlichkeitswerte zu hoch, sodass es sinnvoll ist die Berechnung von Ähnlichkeitswerten auf bestimmte Kandidaten zu begrenzen, womit wir uns jedoch nicht näher befasst haben. Desweiteren ist eine Normalisierung der Ähnlichkeitswerte notwendig, da sich Ähnlichkeitswerte zwischen Pseudosynsets verschiedener Ressourcen nicht ohne weiteres miteinander vergleichen lassen. Diesbezüglich haben wir eine Methode vorgestellt, welche Ähnlichkeitswerte anhand des durchschnittlichen Ähnlichkeitswerts zwischen den Pseudosynsets von zwei Ressourcen normalisiert.

Der korrektive Ansatz baut hingegen auf bereits vorhandenen paarweisen Alignments auf, indem aus diesen ein ungerichteter, ungewichteter Graph erzeugt wird. Aufgrund der Struktur dieses Graphen versucht der Ansatz dann fehlende Kanten zu identifizieren und den Graphen durch das Hinzufügen und/oder Entfernen von Kanten zu korrigieren. Eine Beschränkung bei dieser Methode besteht darin, dass nur Kanten innerhalb der vorgegebenen Komponenten korrigiert werden können, wodurch die Wirksamkeit des Ansatzes eingeschränkt ist.

Die Evaluation der Ansätze hat gezeigt, dass beide Ansätze das Potenzial besitzen, paarweise Alignments bezüglich ihrer Qualität zu übertreffen. Bei dem konstruktiven Ansatz erreichen wir mit dem Hierarchisch Agglomerativen Clustering je nach Gold-Standard F-Measure Werte bis zu 65,8%, mit Topological Overlap und Newman Clustering bis zu 64,0%. Dies bedeutet eine Verbesserung um bis zu 16% im Vergleich zum paarweisen Alignment. Es ist jedoch zu beachten, dass die Verfahren auf den verschiedenen Gold-Standards unterschiedlich gut arbeiten, wodurch die Verbesserung zum Teil nur sehr gering oder sogar negativ ausfällt. Insgesamt erkennen wir jedoch eine Qualitätsverbesserung. Bei dem korrektiven Ansatz beträgt die Verbesserung der F-Measure Werte im Vergleich zu den Ausgangsdaten bei den Komponenten auf denen er angewendet werden kann 1 bis 5%.

Die beobachtete Qualitätsverbesserung hängt insbesondere mit der globalen Struktur zusammen, welche sich durch die Betrachtung von mehr als zwei Ressourcen ergibt. Während bei einem paarweisen Alignment jede Kante isoliert (lokal) betrachtet und klassifiziert wird, können wir in einem multiplen Alignment anhand der globalen Struktur entscheiden, ob zwei Pseudosynsets einander zugewiesen werden oder nicht. So können zwei Pseudosynsets trotz eines niedrigen Ähnlichkeitswerts einander zugewiesen werden, wenn sie beispielsweise beide hohe Ähnlichkeitswerte zu einem dritten Pseudosynset haben.

Um jedoch das volle Potenzial multipler Alignments auszuschöpfen ist es wichtig weiter an der Lösung oben besprochener Herausforderungen, insbesondere der Vergleichbarkeit von Ähnlichkeitswerten und der Kandidatenextraktion, zu arbeiten. Außerdem sollten für die Berechnung der paarweisen Ähnlichkeitswerte bessere Ansätze als das einfache String-basierte Verfahren genutzt werden, sofern wirklich qualitativ hochwertige multiple Alignments das Ziel sind und nicht lediglich gezeigt werden soll, welche Vorteile multiple Alignments gegenüber paarweisen Alignments haben (wie in dieser Arbeit).

Nachdem diese Arbeit nun das Potenzial und die Probleme multipler Alignments in Bezug auf paarweise Alignments aufgezeigt hat, ist für die Zukunft eine Optimierung der Verfahren für multiple Alignments an sich erstrebenswert. Neben einer Behandlung der oben besprochenen Herausforderungen wären alternative Evaluationsmethoden denkbar. So könnte beispielsweise ein Gold-Standard speziell für multiple Alignments erstellt werden. Desweiteren könnte die Einbeziehung weiterer lexikalisch-semantischer Ressourcen wie FrameNet oder VerbNet in das multiple Alignment von Interesse sein. Diese Ressourcen unterscheiden sich in ihrem Aufbau deutlich von den hier genutzten Ressourcen, was nochmals völlig neue Fragestellungen und Probleme aufwerfen könnte.

8 Glossar

8.1 Begriffe

- **Lexikalische Ressource**: Eine lexikalische Ressource besteht aus einer Auflistung von Wörtern und deren möglichen Bedeutungen.
- **Sense**: Ein Sense ist durch ein Wort und dessen zugehörige Bedeutung identifiziert. Ein Wort kann verschiedene Senses haben (z.B. Bank im Sinne von Geldinstitut und Bank im Sinne von Sitzbank). Zwei oder mehr verschiedene Wörter stellen immer auch verschiedene Senses dar auch wenn sie die gleiche Bedeutung haben (z.B. Auto und Automobil).
- **Synset**: Der Begriff Synset wurde im Kontext der lexikalisch-semantischen Ressource WordNet erschaffen. Ein Synset („set of synonyms“) fasst alle synonymen Wörter einer Ressource (d.h. Wörter, die eine bestimmte Bedeutung teilen) zusammen. Unterschiedliche Wörter, die jedoch die gleiche Bedeutung beschreiben (z.B. Auto und Automobil), bilden folglich ein Synset. Jedes zu einem Synset gehörende Synonymwort stellt einen eigenen Sense dar.
- **Synonymie**: Zwei oder mehr Senses/Synsets werden als synonym bezeichnet, wenn die zugehörigen Wörter die gleiche Bedeutung haben. Während innerhalb einer lexikalischen Ressource bereits alle Synonymwörter in einem einzigen Synset zusammengefasst sind, können Synsets aus unterschiedlichen lexikalischen Ressourcen die gleiche Bedeutung beschreiben und somit synonym sein. Senses können auch innerhalb einer lexikalischen Ressource synonym sein, da jedes Synonymwort einen eigenen Sense darstellt. Wir betrachten Synonymie als reflexiv, symmetrisch und transitiv.
  - Ein Sense/Synset A ist synonym zu sich selbst.
  - Wenn Sense/Synset A synonym zu Sense/Synset B ist, dann ist auch Sense/Synset B synonym zu Sense/Synset A.
  - Wenn Sense/Synset A synonym zu Sense/Synset B ist und Sense/Synset B synonym zu Sense/Synset C, dann ist Sense/Synset A synonym zu Sense/Synset C.
- **Paarweises/multiples Alignment**: Unter einem Alignment verstehen wir die Zuweisung von synonymen Senses/Synsets aus genau zwei (paarweises Alignment) bzw. mehr als zwei (multiples Alignment) lexikalischen Ressourcen.
- **Komponente**: Eine Komponente in einem Graphen stellt einen Subgraphen dar, in dem jeder Knoten über einen Pfad mit allen anderen Knoten der Komponente verbunden ist (zusammenhängend).
- **Isolierte Komponente**: Bezeichnet eine Komponente, die aus nur einem Knoten besteht, der keine Kante zu anderen Knoten besitzt.
- **Split**: Ein Split teilt eine Komponente durch Entfernen von Kanten in zwei oder mehr Komponenten (Kantenschnitt).
- **No-Split Option** (Korrektiver Ansatz): Die Option bei der eine als fehlerhaft identifizierte Komponente alleine durch das Hinzufügen von fehlenden Kanten korrigiert wird.

- **Split Option** (Korrektiver Ansatz): Eine Option bei der eine Komponente durch das Entfernen bestimmter Kanten in zwei Komponenten zerteilt wird.
- **Cluster**: Ein stark zusammenhängender Teilgraph einer Komponente.

8.2 Abkürzungen

- **WN**: WordNet
- **WP**: Wikipedia
- **WKT**: Wiktionary
- **OW**: OmegaWiki
- **TP**: True Positive (korrekterweise aligniert)
- **TN**: True Negative (korrekterweise nicht aligniert)
- **FP**: False Positive (fälschlicherweise aligniert)
- **FN**: False Negative (fälschlicherweise nicht aligniert)

8.3 Formeln

- **Precision**: $\frac{TP}{TP+FP}$
- **Recall**: $\frac{TP}{TP+FN}$
- **Accuracy**: $\frac{TP+TN}{TP+FP+TN+FN}$
- **F-Measure**: $\frac{2 \cdot precision \cdot recall}{precision+recall} = \frac{2 \cdot TP}{2 \cdot TP+FP+FN}$

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Agirre and Soroa, 2009] E. Agirre and A. Soroa. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL)*, pages 33–41, Athens, Greece, 2009.

[Avrachenkov *et al.*, 2008] K. Avrachenkov, V. Dobrynin, D. Nemirovsky, S. K. Pham, and E. Smirnova. Pagerank based clustering of hypertext document collections. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, Special Interest Group on Information Retrieval 2008, pages 873–874, New York, NY, USA, 2008. Association for Computing Machinery.

[Baker and Fellbaum, 2009] C. F. Baker and C. Fellbaum. WordNet and FrameNet as complementary resources for annotation. In *Proceedings of the Third Linguistic Annotation Workshop*, Association for Computational Linguistics - International Joint Conference on Natural Language Processing 2009, pages 125–129, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Brandes, 2001] U. Brandes. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[de Melo and Weikum, 2010] G. de Melo and G. Weikum. Providing Multilingual, Multimodal Answers to Lexical Database Queries. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 348–355, Valletta, Malta, 2010.

[Euzenat and Shvaiko, 2007] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[Fellbaum, 1998] C. Fellbaum. WordNet: An electronic lexical database. The MIT Press, 1998.

[Feng and Doolittle, 1987] D. Feng and R. Doolittle. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.

[Giles, 2005] J. Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–1, 2005.

[Gurevych *et al.*, 2012] I. Gurevych, J. Eckle-Kohler, S. Hartmann, M. Matuschek, C. M. Meyer, and C. Wirth. Uby - A Large-Scale Unified Lexical-Semantic Resource. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, number 13, pages 580–590, April 2012.

[Hansen, 2004] A. Hansen. *Bioinformatik : Ein Leitfaden für Naturwissenschaftler*. Birkhäuser, 2. überarb. und erweiterte Auflage edition, 2004.

[Haveliwala, 2003] T. H. Haveliwala. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15:784–796, 2003.

[Ide and Wilks, 2007] N. Ide and Y. Wilks. Making Sense about Sense. In *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*, pages 47–74, 2007.

[Jain *et al.*, 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computint Surveys*, 31(3):264–323, September 1999.

[Johansson and Nugues, 2007] R. Johansson and P. Nugues. Using WordNet to extend FrameNet coverage. In *Proceedings of the Workshop on Building Frame-semantic Resources for Scandinavian and Baltic Languages*, pages 27–30, Tartu, Estonia, 2007.

[Jurafsky and Martin, 2000] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2000.

[King, 1967] B. King. Step-Wise Clustering Procedures. *Journal of the American Statistical Association*, 62(317):86–101, 1967.

[Li and Horvath, 2007] A. Li and S. Horvath. Network neighborhood analysis with the multi-node topological overlap measure. *Bioinformatics*, 23(2):222–231, 2007.

[Meyer and Gurevych, 2010] C. M. Meyer and I. Gurevych. How Web Communities Analyze Human Language: Word Senses in Wiktionary. In *Proceedings of the Second Web Science Conference*, Raleigh, NC, USA, 2010.

[Meyer and Gurevych, 2011] C. M. Meyer and I. Gurevych. What Psycholinguists Know About Chemistry: Aligning Wiktionary and WordNet for Increased Domain Coverage. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 883–892, Chiang Mai, Thailand, November 2011.

[Mihalcea *et al.*, 2006] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2006.

[Mihalcea, 2007] R. Mihalcea. Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Asociation for Computational Linguistics*, Rochester, NY, USA, 2007.

[Newman and Girvan, 2004] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, February 2004.

[Newman, 2004] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):9, 2004.

[Niemann and Gurevych, 2011] E. Niemann and I. Gurevych. The People's Web meets Linguistic Knowledge: Automatic Sense Alignment of Wikipedia and WordNet. In *Proceedings of the 9th International Conference on Computational Semantics*, pages 205–214, Singapore, January 2011.

[Page *et al.*, 1999] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[Ponzetto and Navigli, 2010] S. P. Ponzetto and R. Navigli. Knowledge-rich Word Sense Disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics 2010, pages 1522–1531, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[Potthast *et al.*, 2008] M. Potthast, B. Stein, and M. Anderka. A Wikipedia-Based Multilingual Retrieval Model. In *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 522–530. Springer, 2008.

[Raghavan *et al.*, 2007] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76:036106, September 2007.

[Ravasz *et al.*, 2002] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi. Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297(5586):1551–1555, 2002.

[Ruiz-Casado *et al.*, 2005] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets. In PiotrS. Szczepaniak, Janusz Kacprzyk, and Adam Niewiadomski, editors, *Advances in Web Intelligence*, volume 3528 of *Lecture Notes in Computer Science*, pages 380–386. Springer, 2005.

[Shi and Mihalcea, 2005] L. Shi and R. Mihalcea. Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 3406 of *Lecture Notes in Computer Science*, pages 100–111. Springer, 2005.

[Suchanek *et al.*, 2007] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW 2007, pages 697–706. Association for Computing Machinery, 2007.

[Tittmann, 2003] P. Tittmann. *Graphentheorie - Eine anwendungsorientierte Einführung*. Carl Hanser Verlag, 2003.

[Toral *et al.*, 2009] A. Toral, O. Ferrandez, E. Agirre, and R. Munoz. A study on Linking Wikipedia categories to Wordnet using text similarity. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, 2009.

[Wolf and Gurevych, 2010] E. Wolf and I. Gurevych. Aligning Sense Inventories in Wikipedia and WordNet. In *Proceedings of the First Workshop on Automated Knowledge Base Constructions (AKBC)*, 2010.

[Zhang and Bodenreider, 2005] S. Zhang and O. Bodenreider. Alignment of multiple ontologies of anatomy: deriving indirect mappings from direct mappings to a reference. *AMIA Annual Symposium proceedings AMIA Symposium AMIA Symposium*, 2005:864–868, 2005.