

IMS

Bachelorarbeit

**Die Rolle von Abstraktheit und  
Wortbedeutungen bei der  
Unterscheidung von wörtlichen und  
nicht-wörtlichen Bedeutungen von  
deutschen Partikelverben**

Jan Gemander

<b>Studiengang:</b>	Informatik
<b>Prüfer/in:</b>	Prof. Dr. Sabine Schulte im Walde
<b>Betreuer/in:</b>	Prof. Dr. Sabine Schulte im Walde, Maximilian Köper, M.Sc.
<b>Beginn am:</b>	1. Dezember 2017
<b>Beendet am:</b>	1. Juni 2018



## **Kurzfassung**

Ziel dieser Arbeit ist es, die wörtliche bzw. metaphorische Verwendung von deutschen Partikelverben richtig identifizieren zu können. Als erster Schritt werden hier der von Köper und Schulte im Walde[KW16] vorgeschlagene Klassifizierer in Python reimplementiert und die Menge der verwendeten Daten erweitert. Als zweiten zentralen Schritt dieser Arbeit variieren wir die verwendeten Abstraktheitsfeatures und experimentieren mit eigens erzeugten Werten. Als letztes Feature werfen wir einen Blick auf die Verwendung der Wortbedeutungen in Form der im Duden definierten Senses, welche wir mithilfe des Lesk-Algorithmus[Les86] sowie Wortvektoren bestimmen.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung/Motivation</b>	<b>11</b>
<b>2</b>	<b>Verwandte Arbeiten</b>	<b>13</b>
2.1	Klassifizierung von Wörtlich und Metaphorisch . . . . .	13
2.2	Verwendung von Abstraktheitswerten . . . . .	13
<b>3</b>	<b>Datensätze</b>	<b>15</b>
<b>4</b>	<b>Features</b>	<b>17</b>
4.1	Distributionelle Semantik . . . . .	17
4.2	Abstraktheitswerte . . . . .	19
4.3	Duden-Senses . . . . .	21
<b>5</b>	<b>Experimente</b>	<b>23</b>
5.1	Vergleich verschiedener Klassifizierungen . . . . .	23
5.2	Abstraktheit . . . . .	24
5.3	Anwendung auf den weiteren Datensätzen . . . . .	25
5.4	Duden-Sense . . . . .	25
<b>6</b>	<b>Qualitative Analyse</b>	<b>29</b>
6.1	Annotierte Werte . . . . .	29
6.2	Partikel . . . . .	30
6.3	Abstraktheitswerte . . . . .	31
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>33</b>
	<b>Literaturverzeichnis</b>	<b>35</b>



# Abbildungsverzeichnis

6.1	Partikel-Verteilung . . . . .	30
6.2	Vergleich der eigenen Abstraktheitswerte mit denen in Affektive Normen . . . . .	31





# Tabellenverzeichnis

3.1	Übersicht der Datensätze . . . . .	16
5.1	Ergebnisse der beschriebenen Features, 3: Cluster-Tag anstatt Nomen 5: Abstraktheitswerte der Nomen als einziges Feature, 6: Abstraktheitswerte für Cluster der Nomen . . . . .	24
5.2	Ergebnisse zu den in Abschnitt 4.2.1 mit gleicher Nummerierung beschriebenen Features, #Betrachtet steht für die Anzahl der konkretesten und abstraktesten Wörter der Referenzliste, Sim für die Kosinus-Ähnlichkeit des jeweils Ähnlichsten . . . . .	24
5.3	Ergebnisse der interessanten Features auf den verschiedenen Datensätzen, Kombination enthält alle 3 darüber genannten Features, Alle ist die Summe der 3 Datensätze . . . . .	26
5.4	Ergebnisse der betrachteten Sense Varianten . . . . .	27
6.1	Ergebnisse zu annotierten N-Lit Werten, Klammern stellen Schranken dar, wobei eckige inklusive und runde ohne den gegebenen Wert sind . . . . .	29
6.2	Ergebnisse pro Partikel . . . . .	30



# 1 Einleitung/Motivation

Während der Mensch intuitiv Metaphern in seiner Sprache verwendet, ist es für ein Programm schwer eigenständig zu erkennen, welche Bedeutung eines Wortes verwendet wird. Dies ist jedoch in vielen Fällen notwendig um einen Satz richtig zu verstehen. Im Bereich der Computerlinguistik stellt dies einen zentralen Punkt dar um den Text korrekt weiter zu verarbeiten, weshalb bereits ein beträchtlicher Anteil an Arbeit auf diesem Gebiet getan wurde [BS06; BS07; KW16; SL09; SS13; TNAC11].

Zentralpunkt dieser Arbeit ist es Partikelverben, welche durch die Zusammensetzung von Partikel (auf, ab, an . . . ) und Basisverb (keimen, machen, laufen . . . ) und den damit häufig kommenden Bedeutungsveränderungen einen eher schwereren Teil der natürlichen Sprachverarbeitung darstellen, richtig in wörtlich sowie metaphorisch klassifizieren können. So hat beispielsweise die Phrase *Hoffnung keimt auf* nur wenig mit der eigentlichen Bedeutung von keimen zu tun.

Grundstein dieser Arbeit bildet die zuvor gegangene Arbeit von Köper & Schulte im Walde [KW16] welche einen Klassifizierer sowie verschiedene Features vorstellen und damit eine Genauigkeit von 86,8% erreichen. Der erste Schritt dieser Arbeit ist es die einen Klassifizierer mit diesen Features in Python zu reimplementieren.

Ein Kernpunkt dieser Arbeit ist außerdem der Zusammenhang von Abstraktheit und der metaphorischen Anwendung von Wörtern. Annahme ist, dass ein Partikelverb mit einem Abstrakten Kontext wie in *Hoffnung keimt auf* in der Regel metaphorisch verwendet wird [TNAC11] . Hierbei werden verschiedene Methoden betrachtet Abstraktheitswerte zu erzeugen. Ein weiterer Gedanke den wir in dieser Arbeit verfolgen werden ist, das Ergebnis eines Disambiguierungs-Problems als Feature zu verwenden. Dabei identifizieren wir für Partikelverben die dazugehörige Bedeutung im Duden, und extrahieren daraus Features.

Dabei versuchen wir außerdem die Relevanz der Ergebnisse zu erhöhen, indem wir weitere Datensätze betrachten, um Probleme des Klassifizierers zu identifizieren.



## 2 Verwandte Arbeiten

Die für diese Ausarbeitung relevanten Arbeiten beziehen sich auf die Verwendung bzw. Identifizierung von metaphorischer Sprache, sowie den damit zusammenhängenden Abstraktheitsgrad des Kontextes.

### 2.1 Klassifizierung von Wörtlich und Metaphorisch

Birke und Sakar [BS06; BS07] betrachten das Thema als ein Disambiguierungs-Problem<sup>1</sup>, mit jeweils den zwei Bedeutungen Wörtlich und Metaphorisch für die verwendeten Worte. Für die Klassifizierung der dabei betrachteten englischen Verben werden vorher annotierte Mengen an Seed-Sätzen verwendet. Die größte Ähnlichkeit des Satzes mit dem betrachteten Verb im Vergleich zu den dazugehörigen Seed-Sätzen bestimmt die Bedeutung.

Sporleder und Li [SL09] beschränken sich auf die Bestimmung der Verben von in Sätzen verwendeten Redewendungen. Hierbei verwenden sie zunächst Zusammenhangs-Ketten, in denen die Komponenten des Sprichwortes mit denen des Satzes zusammen in Ketten von Wörtern mit verwandten Bedeutungen gesammelt werden, deren Länge als Feature verwendet wird. Als zweites schlagen sie einen Zusammenhangs-Graph mit den Komponenten als Knoten und der zwischen den Wörtern bestehenden Zusammengehörigkeit als Kanten vor. So gehen sie davon aus, dass falls die durchschnittliche Zusammengehörigkeit steigt, nachdem die Wörter der Redewendung entfernt wurden, das Verb metaphorisch verwendet wurde.

Shutova et al. [SS13] trainiert auf der Basis von Links zwischen Verb- und Nomen-Clustern. Dafür wird eine Menge von metaphorischen Sätzen benötigt, aus denen solch eine Abbildung erzeugt werden kann. Verben in weiteren Sätzen werden anhand der vorliegenden Links zu den im Kontext liegenden Nomen klassifiziert.

Broadwell et al. [BBC+13] schlagen vor, dass Metaphern leicht vorstellbare Wörter sind, welche weder semantisch noch thematisch in den Kontext passen. Die Implementierung auf verschiedenen Sprachen identifiziert zunächst die leicht vorstellbaren Wörter des ausgewählten Abschnittes und schließt im Anschluss nicht brauchbare mithilfe von thematischen Zusammenhangs-Ketten und Semantischen-Clustering aus.

### 2.2 Verwendung von Abstraktheitswerten

Bereits 1980 stellten Lakoff und Johnsen [LJ80] die Behauptung auf, dass durch die Verwendung von Metaphern die Sprache abstrakter wird. Darauf aufbauend schlagen Turney et al. [TNAC11] die Hypothese auf, dass der Abstraktheitsgrad im Kontext des verwendeten Wortes in Relation

---

<sup>1</sup>Identifizierung der verwendeten Wortbedeutung

zur Wahrscheinlichkeit der metaphorischen Verwendung liegt. Die Features erhalten sie aus den Abstraktheitswerten der Kontextwörter zu den betrachteten Adjektiv+Nomen Phrasen sowie Verben. Tsvetkov et al. [TMG13] verwenden neben dem Abstraktheitsgrad die für Verben und Nomen in Wordnet gegebenen semantischen Kategorien sowie boolesche Werte für Person, Organisation und Ort als Features. Dieses Modell verwenden sie um zunächst englische Verben korrekt klassifizieren zu können, sind aber dazu in der Lage das System auf beliebige Sprachen erweitern. Dieses Modell verbessern sie 2014 [TBG+14], indem zusätzlich die Bildbarkeit und gelernte Wortvektoren als Features und die Möglichkeit, Adjektiv-Nomen Phrasen zu klassifizieren, eingeführt werden. Körper und Schulte im Walde [KW16] spezialisieren sich auf das Klassifizieren von Partikelverben. Neben Abstraktheit und Bildbarkeit werden als Features die Frequenz der vorkommenden Wörter, verbessert durch Nomen-Cluster, sowie der Vergleich des semantischen Zusammenhalts von Partikelverb bzw. Basisverb im gegebenen Kontext verwendet.

## 3 Datensätze

Die im Rahmen dieser Arbeit ausgeführten Experimente finden auf insgesamt drei Datensätzen, bestehend aus einzelnen Sätzen mit Partikelverben, statt. Dazugehörigen Sätze werden auf Basis der metaphorischen Verwendung des Partikelverbs kategorisiert:

1. LITERAL: Der Pflanzensamen keimte auf.
2. NON-LITERAK: Hoffnung keimte in den Menschen auf.

Jeder dieser Sätze wurde von Menschen mit einem Computer-linguistischen Hintergrund darauf hin bewertet worden, ob das Partikelverb seine Bedeutung beibehält(1) oder im metaphorischen Sinne(2) verwendet wurde. Die Bewertung beruht auf einer Skala von 0 (wörtlich/Lit) bis 5 (nicht-wörtlich/N-Lit).

Zusätzlich wird der TreeTagger des CIS-München [Sch95] verwendet um Part-of-Speech(POS) Informationen sowie Lemma der jeweiligen Sätze zu erhalten. Mithilfe der POS Informationen werden in Sätzen mit voneinander getrennten Verben und Partikeln die dazugehörigen Lemmata für die weitere Klassifizierung zusammengefügt (*Der Betrug fliegt auf* wird zu *der Betrug auffliegen*). Insofern diese Heuristik für einen Satz nicht genau ein Partikelverb, welches in einer vordefinierten Liste enthalten ist, identifizieren kann, wird der Satz ignoriert. Eine allgemeine Übersicht zu den Charakteristiken der im folgenden beschriebenen Datensätze findet sich in Tabelle 3.1.

### 3.0.1 NAACL

Der größte für die Klassifizierung verwendete Datensatz, bestehend aus 6436 annotierten Sätzen wird aus vorhergehenden, bei der *North American Chapter of the Association for Computational Linguistics* (NAACL) veröffentlichten Arbeit von Körper & Schulte im Walde[KW16] übernommen. Die darin enthaltenen Sätze wurden für die zuvor ausgewählten Partikelverben mit den Partikeln {ab, an, auf, aus, durch, ein, mit, nach, vor, zu} aus den Deutschen Web-Korpus *DECOW14AX* entnommen.

Von diesen Sätzen werden in Folge der zuvor beschriebenen Heuristik 6360 Sätze verwendet.

### 3.0.2 Germanet

Der Germanet-Datensatz besteht aus an der Universität Stuttgart gesammelten annotierten Sätzen, welche aus GermaNet extrahiert wurden. GermaNet ist eine Datenbank mit deutsche Nomen, Verben und Adjektive, die jeweils genauer kategorisiert und semantisch verknüpft werden. Für jede Bedeutung der Partikelverben sollten also alle Bedeutungen enthalten sein.

Dabei wurden nur Partikelverben mit einem Partikel  $\in$  {ab, an, auf, aus} verwendet. In Tabelle

### 3 Datensätze

---

Dataset	#Partikel	#Partikelverben	Baseline	#Lit	#N-Lit	#Sätze
Naacl	10	165	0,650	4136	2224	6360
Germanet	4	944	0,857	1475	247	1722
Spatial	8	82	0,650	739	398	1137
Alle	11	1024	0,689	6350	2869	9119

**Tabelle 3.1:** Übersicht der Datensätze

3.1 ist sichtbar, dass dies der einzige Datensatz mit einer deutlich größeren Menge an wörtlichen Partikelverb-Sätze in Relation zu den nicht-wörtlichen ist. Dadurch erreicht bereits ein trivialer Klassifizierer eine hohe Genauigkeit. Hiervon werden 1722 von anfänglich 1911 Sätzen weiterverwendet.

#### 3.0.3 Spatial

Der letzte Datensatz auf dem die Klassifizierungen durchgeführt werden ist der Spatial-Datensatz. Die hierin enthaltenen Sätze wurden für Experimente zu den räumlichen Richtungen der Sätze von Diego Frassinelli gesammelt. Darin sind 88 Partikelverben mit jeweils ca. 15 dazugehörigen Sätzen enthalten. Die für die Partikelverben verwendeten Partikel beschränken sich auf {an, auf, bei, ein, mit, nach, vor, zu}. Zusätzlich sind in dem Datensatz Sätze ohne Partikel vorhanden, welche für weitere Experimente ausgelassen werden. Die Menge der enthaltenden Sätze beläuft sich auf 2680, von denen nur 1137 in den Experimenten verwendet werden.



## 4 Features

In diesem Kapitel wollen wir verschiedene Features vorstellen, welche wir später für im Training verwenden wollen. Dabei werden wir aus der Arbeit von Körper und Schulte im Walde[KW16] die Features der Wortfrequenz und des Distributionellen Fits reimplementieren, und unseren Fokus auf die Implementierung verschiedener Abstraktheitswerte fokussieren. Des Weiteren versuchen wir die Wortbedeutungen als Feature einzubringen.

### 4.1 Distributionelle Semantik

#### 4.1.1 Unigramme

Als erstes intuitives Feature verwenden wir für jeden Satz die im Partikelwort-Kontext auftretenden Wörter. Diese werden als eine Art bag-of-words Modell, also unabhängig von der Position des Wortes verwendet. Der Featureerraum beträgt hier also die Größe des Vokabulars. Dem jeweiligen Partikelverb wird die Kategorie mithilfe der Kontextwörter-Wahrscheinlichkeit im Trainingset in Bezug auf die Klassen {Lit, N-Lit} zugewiesen. Die Motivation ist, dass für einige, insbesondere stark konkrete sowie abstrakte Wörter, klar differenzierbar ist, wie die damit in Verbindung auftretenden Partikelverben zu klassifizieren sind.

Für einen Satz *Hoffnung aufkeimen Menschen* werden die jeweiligen Wahrscheinlichkeiten von *Hoffnung* und *Menschen* in Bezug auf wörtlich und metaphorisch multipliziert. Falls gilt, dass  $P(N-Lit|Hoffnung, Mensch) > P(Lit|Hoffnung, Mensch)$  wird *aufkeimen* als metaphorisch ausgegeben, sonst als wörtlich.

Für das Training wird der Naive-Bayes<sup>1</sup> Klassifizierer von Scikit-Learn in Bezug auf die Wortfrequenzen verwendet.

#### 4.1.2 Distributionelles Fit

Eine weitere Idee bezieht sich darauf, dass Partikelverben, die im Kontext zu Wörtern stehen, welche normalerweise nicht mit dem Basisverb auftreten, sich die Bedeutungen unterscheiden. So ist Grundgedanke der distributionellen Semantik, dass Wörter welche immer im gleichen Kontext auftreten, austauschbar seien. So würden wir in dem Satz *Den Job kannst du dir abschminken*, nach dem Ersetzen des Partikelverbs durch dessen Basisverb *Den Job kannst du dir schminken* erhalten, ein Satz der kaum Sinn ergibt. Hierfür wollen wir berechnen, wie sehr Partikelverb sowie Basisverb in den Kontext des Partikelverbs passen. Dafür berechnen wir die durchschnittliche Ähnlichkeit, von Partikelverb bzw. Basisverb in Bezug auf die Kontextwörter.

---

<sup>1</sup>[http://scikit-learn.org/stable/modules/naive\\_bayes.html](http://scikit-learn.org/stable/modules/naive_bayes.html)

**Erzeugung von Wortvektoren** Eine Möglichkeit die Ähnlichkeit von Worten sowie die Wahrscheinlichkeitsverteilung in Sätzen berechnen zu können, sind Vektormodelle. Im Folgenden wollen wir beschreiben, wie wir Wortvektoren, genauer eine Wort-Kontext Matrix, für den späteren Gebrauch erzeugen können. Die Matrix soll für jedes Zielwort aus unserer Datenmenge die im Kontext auftretenden Wörter zählen. Die Idee dahinter ist, dass Wörter, welche im gleichen Kontext auftreten, meist ähnliche Bedeutungen haben (Harris, 1954).

Da die gegebene Datenmenge zu klein ist um relevante Wortvektoren zu erhalten, verwenden wir den DECOW16 AX Datensatz mit ca. 600 Millionen annotierten Sätzen als Korpus. Für jedes der in einem Satz auftretenden Zielwörter betrachten wir nun die Adjektive, Adverbien, Nomen sowie Verben, welche in einem Fenster von 20, also 10 Wörter zur linken sowie 10 zur rechten des Zielwortes. Jedes der auftretenden Kontextwörter erhöht dann einen Zähler  $Counter[Zielwort][Kontextwort]$  um eins. In unserer Implementierung werden pro Satz für ein Zielwort der Zähler von genau 20 Wörtern erhöht insofern die Satzgrenzen keine Beschränkung darstellen, nicht interessante Kontextwörter werden also auch für die Fenster ignoriert. Sowohl für Zielwörter als auch Kontextwörter werden die Lemmata der Wörter betrachtet.

Sei für das Wort Baum 1.200 Mal das Wort Vogel im Kontext (z.B. im Satz *Der Vogel saß auf dem Ast des Baumes*). Dann gilt  $Counter[Baum][Vogel] = Counter[Vogel][Baum] = 1200$ .

Im Fall von Verben welche gesplittet von Partikeln im Satz auftreten ist im Korpus die Information enthalten zu welchem Verb ein Partikel gehört, sodass wir diese zusammenführen können, bevor wir die Kontextwörter zählen. Falls in dem Lemma eines Wortes mehrere Worte aufgezählt sind, so zählen wir jedes dieser Worte als Kontextwort. Um unnötige Daten auszuschließen werden für Dimensionen alle Kontextwörter, welche im Korpus mindestens 50-Mal auftreten, für die Vektoren verwendet. Ausgegeben wird die Matrix in einem HDF5-Format, wobei die Zielwörter mit ihren Vektoren als Daten auf die Menge der Datasets abgebildet werden.

**Positive-Pointwise-Mutual-Information (PPMI)** Jedoch ist die Frequenz von Worten als Maß nicht gerade optimal, da durch die verschiedene Häufigkeit von Wörtern die Aussagekraft über das gemeinsame Auftreten zweier Wörter beschränkt ist [JM14]. Um einen gemeinsamen Informationsgehalt von einem Zielwort  $w$  und Kontextwort  $c$  zu erhalten kann PMI verwendet werden:

$$PMI(w, c) = \log_2\left(\frac{P(w, c)}{P(w) * P(c)}\right)$$

PMI liefert die Information wie oft ein Zielwort und Kontextwort in Relation zu ihrer eigenen Häufigkeit zusammen auftreten. Ein positiver Wert wird erreicht, wenn  $w$  und  $c$  öfters zusammen im Korpus auftauchen als individuell. Die Erweiterung  $PPMI(w, c) = \max(PMI(w, c), 0)$  setzt den Wert für Wörter welche oft separat voneinander auftreten auf 0.

**Singulärwertzerlegung (SVD)** Eine weitere Möglichkeit die Vektoren zu verbessern stellt SVD dar. Dieses ermöglicht uns die Menge an Dimensionen, also in unserem Fall die Anzahl an Kontextwörtern, auf die wichtigsten zu reduzieren. Neben der dadurch eingesparten Zeit, werden also auch Kontextwörter, welche in einem Vergleich, durch ähnlichen auftreten auf allen Zielwörtern, ungewollt normalisieren würden.

Hierfür wurde die Implementierung TruncatedSVD<sup>2</sup> von Scikit-learn verwendet um die Länge der Vektoren in 7 Iterationen von in etwa 700.000 auf 1.000 zusammen zu fassen. Für alle folgenden Experimente werden diese Vektoren verwendet.

**Clustering** Um die Informationen von selten vorkommenden Worten zu generalisieren können wir Clustering verwenden. Mithilfe der eben bestimmten Vektoren werden unsere Nomen von der Scikit-Learn Implementierung für KMean Clustering<sup>3</sup> in 750 Cluster aufgeteilt. Indem wir später alle Nomen durch ihre Cluster-Tags ersetzen hoffen wir bessere Ergebnisse zu erhalten, da die Worte in diesem Vorbereitungsschritt bereits bezüglich ihrem Kontext genauer kategorisiert werden.

**Distributionelles Fit** Mithilfe dieses Features wollen wir übergeben, wie sehr sich die Bedeutung des verwendeten Partikelverbs und die des Basisverbs unterscheiden. In unserem Beispielsatz *Den Job kannst du dir abschminken*, bzw. *Den Job kannst du dir schminken* sehen wir wie wichtig ein Partikel für die Bedeutung des Satzes sein kann.

Für dieses Feature berechnen wir die Kosinus-Ähnlichkeit von jeweils Partikelverb und Basisverb zu den im Satzkontext auftretenden:

1. Nomen
2. Verben
3. Adjektive
4. Adverbien

Dem Klassifizierer werden für jeweils Partikel- sowie Basisverb 4, insgesamt also 8 Features übergeben. Bei einer kleinen Differenz von Partikel- und Basisverb sind diese austauschbar, verwenden also wahrscheinlich die gleiche Bedeutung.

## 4.2 Abstraktheitswerte

Um auf der Hypothese von Turney et al.[TNAC11] aufzubauen verwenden wir ebenfalls Abstraktheitswerte der Kontextwörter als weiteres Feature. Dies folgt aus der Annahme von Lakoff und Johnsen[LJ80] dass der Kontext von als Metaphern verwendeten Wörtern dazu tendiert abstrakt zu sein. Anstatt einzelne Worte zu betrachten, folgen wir der Motivation, dass in einem abstrakten Kontext hauptsächlich nicht-wörtlich verwendete Partikelverben aufzufinden sind. So sieht man in unserem Anfangsbeispiel,

1. LITERAL: Der Pflanzensamen keimte auf.
2. NON-LITERAL: Hoffnung keimte in den Menschen auf.

---

<sup>2</sup><http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>

<sup>3</sup><http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

dass im metaphorischen Satz abstrakte Worte wie *Hoffnung* vorkommen. Dementsprechend verwenden wir als zugehöriges Features den durchschnittlichen Abstraktheitswert aller Wörter welche im Satz des jeweiligen Partikelverbs auftreten. Hierbei unterscheiden wir zwischen Nomen, Verben, Adjektiven und Adverbien, werden den Fokus später aber auf Nomen legen.

**Affektive Normen** Die dafür verwendeten Werte stammen aus den von Körper und Schulte gesammelten Affektiven Daten<sup>4</sup>. Die gesammelten Informationen sind darin folgendermaßen aufgebaut:

Word	Abstract-Concrete	Arousal	Imageability	Valence
sein	3.731	4.41	4.347	6.768
in	3.883	3.586	5.41	6.192
ein	4.281	3.738	5.014	7.786
werden	3.704	4.433	4.637	6.431
⋮		⋮		

Darin enthalten sind etwa 350.000, auf einer Skala [0,10] für Abstrakt (0) - Konkret (10), Aufregtheit, Bildbarkeit, bewertete deutsche Wörter. Diese wurden in der vorhergehenden Arbeit von Körper, mithilfe der verteilten Semantik auf Basis der Arbeit von Turney et al. (2011) gelernt. Die für diese Arbeit am interessantesten Werte sind die Abstraktheitswerte, welche als Referenzpunkt dienen.

### 4.2.1 Weitere Abstraktheitswerte

Neben den im vorigen Abschnitt genannten Abstraktheitswerten werden in dieser Arbeit weitere Möglichkeiten betrachtet schnell an vernünftige Werte für Wörter zu gelangen. Als Basis dient ein von Lahl et al(2009) an der Uni Düsseldorf veröffentlichter Satz aus 2654 von Menschen annotierten Nomen.

1. Eine Idee ist es aus allen annotierten Wörtern jeweils einfach den Abstraktheitswert des ähnlichsten zu übernehmen. Diesen erhalten wir, indem wir für das jeweilige Wort indem wir mithilfe unserer Wortvektoren die Kosinus-Ähnlichkeit aller annotierten Worte berechnen.
2. Da abstrakte Wörter normalerweise in Zusammenhang von anderen abstrakten Wörtern auftreten baut unser zweites Konzept darauf auf, anhand einer beschränkten Menge der konkretesten und abstraktesten Wörtern zu entscheiden, zu welcher dieser vereinfacht zwei Kategorien ein betrachtetes Wort gehört.
3. Auf diesen Ideen aufbauend, können wir den Abstraktheitswert auch durch zwei oder mehr Features definieren. Indem wir die Menge an konkreten Wörtern wieder beschränken sind wir dazu in der Lage, innerhalb von kurzer Zeit das ähnlichste der gegebenen Wörter zu bestimmen. Zusätzlich zu dem eigentlichen Abstraktheitswert ist es sinnvoll die Kosinus-Ähnlichkeit zu dem dazugehörigen Wort zu übergeben, so kann ein Algorithmus daraus lernen, wie relevant der gegebene Wert ist.

---

<sup>4</sup>[http://www.ims.uni-stuttgart.de/data/affective\\_norms/](http://www.ims.uni-stuttgart.de/data/affective_norms/)

4. Der letzte Versuch beruht darauf lediglich die Kosinus-Ähnlichkeiten zu dem ähnlichsten abstrakten sowie dem ähnlichsten konkreten Wort abzuspeichern. Hierbei ignorieren wir ähnlich wie im zweiten Konzept den eigentlichen Abstraktheitswert, übergeben aber zwei alternative Werte, welche angeben, wie wahrscheinlich es ist, dass das Wort in einem abstrakten bzw. konkreten Kontext auftritt.

### 4.3 Duden-Senses

Als letztes Feature betrachten wir außerdem die Duden-Senses, also die jeweilige Bedeutung der Partikelverben. Für diesen Datensatz werden zusätzlich alle für die Partikelverben im Duden enthaltenen Bedeutungen benötigt. Da diese nur für den Spatial-Datensatz vorhanden sind, beschränken sich unsere Experimente diesbezüglich auf eine relative kleine Datenmenge. In einer ersten Iteration implementieren wir den Lesk-Algorithmus[Les86], um den Sense zu erzeugen. Dieser betrachtet die Menge der sich überschneidenden Wörter unseres Partikelverb-Satzes mit den zuvor gesammelten Sätzen der verschiedenen Bedeutungen des Partikelverbs. Die Bedeutung mit den meisten Überschneidungen wird als Sense verwendet.

Da die Überschneidung von zwei Sätzen mit begrenzten Wortmengen oft leer sein können, verwenden wir in einer zweiten Iteration die durchschnittliche Wortähnlichkeit für die Worte im Satz unseres Partikelverbs zu den jeweiligen Wörtern der verschiedenen Bedeutungen. Der Sense mit der größten Ähnlichkeit wird ausgewählt und weitergegeben.

In beiden Fällen ist der Sense alleine jedoch kaum aussagekräftig, da die Bedeutungen in keiner Art und Weise zu wörtlich/metaphorisch sortiert wurden. Für jede Bedeutung eines Verbs ist also beides möglich (wobei dennoch anzunehmen ist, dass die erste Bedeutung im Duden der eigentlichen Wortbedeutung entspricht). Die betrachteten Features sehen folgendermaßen aus:

1. Partikelverb\_Sense für einen Naive-Bayes-Klassifizierer
2. durchschnittlicher Abstraktheitswert der in der Bedeutung vorkommenden Wörter

Sei unser Wort wieder *aufkeimen* mit zwei Bedeutungen, von denen die erste wörtlich und die zweite metaphorisch ist (muss nicht, es können mehrere, ohne Relation zu wörtlich/metaphorisch existieren). In einer wörtlichen Verwendung würden wir für die erste Implementierungen *aufkeimen\_1*, und für die zweite einen wahrscheinlich nicht-abstrakten Wert (den Definitionen entsprechend) übergeben.



## 5 Experimente

Dieser Abschnitt beschreibt die Durchführung der Klassifizierungen. Zunächst werden auf dem NAACL Datensatz verschiedene Experimente durchgeführt, sowie die eigenen Abstraktheitswerte mit den vorhandenen verglichen. Anhand dieser Ergebnisse werden interessante Features auf den weiteren Datensätzen betrachtet. Für die Bewertung der Klassifizierungen wird jeweils Kreuzvalidierung verwendet, wobei unsere Datensätze in 10 gleich große Mengen unterteilt werden, von denen in 10 Durchgängen jeweils 9 für das Training und 1 für das Testen verwendet werden. Jegliche Informationen zu den dazugehörigen Partikelverben, werden dabei in den Features ausgelassen. Die dabei betrachteten Werte belaufen sich auf die Genauigkeit ( $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ ), Präzision ( $Precision = \frac{TP}{TP+FP}$ ), sowie das F1-Maß als eines der anerkannten besseren Wertungen. Das F1-Maß wird für sowohl wörtlich als auch metaphorisch ausgegeben, während die Präzision aus dem Durchschnitt, gewichtet mit der Anzahl an relevanten Ergebnissen ( $TP + FP$ ), gebildet wird. Durch die Gewichtung würde Genauigkeit = Trefferquote ( $Recall$ ) gelten, weshalb letztere nicht aufgezählt ist.

### 5.1 Vergleich verschiedener Klassifizierungen

In den Experimenten wird für die Klassifizierung der Features, mit Ausnahme der Unigramme, welche einen Naive-Bayes Klassifizierer verwenden, eine Support-Vector-Machine<sup>1</sup> verwendet. Ein Random-Forest Klassifizierer schnitt in den meisten Fällen leicht schlechter ab. Für die Kombination der Features werden diese konkateniert an den Klassifizierer weitergegeben, wobei im Falle der Unigramme die Wahrscheinlichkeiten für wörtlich/metaphorisch durch den zuvor ausgeführten Naive-Bayes als zwei weitere Features übergeben wird. Um Overfitting der übergebenen Wahrscheinlichkeiten zu vermeiden werden die Daten der Unigramme zuvor in zwei Mengen unterteilt, diese werden beide einmal für das Training verwendet, sodass auf dem anderen Datensatz für die dazugehörigen Prognosen berechnet werden können.

Die Ergebnisse der Klassifizierungen finden sich in der Tabelle 5.1. Die Kombination basiert auf den jeweils besten Ergebnissen. Mit Ausnahme der Kombination konnten die Ergebnisse der vorgehenden Papers in Python reproduziert werden. Die verwendeten Abstraktheitswerte liefern ähnlich gute Ergebnisse. Lediglich das Distributionelle Fit, welches scheinbar für fast alle Werte metaphorisch ausgibt, fällt hier mit einer leidet hier unter einer schlechten Präzision.

---

<sup>1</sup><http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>

	Feature	Accuracy	Precision	Lit F1	Non-Lit F1
1	Baseline	0.65			
2	Unigramme	0,76 (+/- 0,03)	0,75	0,82	0,63
3	Unigramme (Nomen Cluster)	0,77 (+/- 0,04)	0,77	0,83	0,66
4	Abstraktheitswerte	0,74 (+/- 0,02)	0,73	0,82	0,56
5	Abstrakt (nur Nomen)	0,74 (+/- 0,04)	0,73	0,82	0,55
6	Abstrakt (Nomen Cluster)	0,72 (+/- 0,03)	0,71	0,8	0,53
7	Distr. Fit	0,72 (+/- 0,04)	0,47	0,78	0,01
8	Kombination (3,5,7)	0,80 (+/- 0,03)	0,79	0,84	0,69

**Tabelle 5.1:** Ergebnisse der beschriebenen Features, 3: Cluster-Tag anstatt Nomen 5: Abstraktheitswerte der Nomen als einziges Feature, 6: Abstraktheitswerte für Cluster der Nomen

## 5.2 Abstraktheit

Da die verschiedenen in Abschnitt 4.2.1 beschriebenen Implementierungen alle auf Abstraktheitswerten des Kontextes aufbauen, übergeben wir mehrmals dasselbe Feature auf verschiedene Arten. Präzision sowie F1-Maß sind für alle Ergebnisse ähnlich und korrelieren in diesem Abschnitt stark mit der Genauigkeit, weshalb wir im Folgenden lediglich letztere betrachten wollen. Zusätzlich alterieren wir für die beschriebenen Features 2-4 die Menge der betrachteten konkretesten sowie abstraktesten Wörtern. Die Ergebnisse des Klassifizierers werden mit der Abstraktheit der Nomen (5 in Tabelle 5.1) auf den NAACL Daten berechnet werden. So können wir die neuen Werte sofort mit den zuvor bestimmten vergleichen.

Feature	Accuracy			
1 Ähnlichstes (alle)	0.70 (+/- 0.03)			
#Betrachtet	1	10	50	100
2 Binär	0,70(+/- 0,03)	0,71(+/- 0,04)	0,670 (+/- 0,032)	0,67 (+/- 0,03)
3 (Sim, Wert)	0,71 (+/-0,03)	0,71(+/- 0,02)	0,70 (+/-0,04)	0,71 (+/-0,04)
4 (Sim-konkret, -abstrakt)	0,72(+/- 0,02)	0,73(+/- 0,03)	0,72(+/- 0,03)	0,73 (+/-0,02)

**Tabelle 5.2:** Ergebnisse zu den in Abschnitt 4.2.1 mit gleicher Nummerierung beschriebenen Features, #Betrachtet steht für die Anzahl der konkretesten und abstraktesten Wörter der Referenzliste, Sim für die Kosinus-Ähnlichkeit des jeweils Ähnlichsten

Wie wir in der Tabelle 5.2 sehen Können bedeuten eine größere Menge an verglichenen Wörtern nicht immer bessere Werte. So sinkt die binäre Erfolgsrate mit einer größeren Menge, während die anderen relativ konstant bleiben. Die Vermutung liegt nahe, dass durch den Zusammenhalt von abstrakten Wörtern jeweils bereits die Tendenz ausreicht (Binär), wobei der Übergang durch die Verwendung mehrerer weniger extremen (abstrakten bzw. konkreten) Wörtern undeutlicher wird. Ebenfalls ist interessant, dass die eigentliche Abstraktheitswert keine Verbesserung liefert. Im Gegenteil ist der binäre Wert fast so gut wie die Auswahl des ähnlichsten Wertes, hierbei spielt aber auch die Extrapolation eine Rolle, dadurch, dass für Worte, welche kein passendes Gegenstück im Datensatz enthalten, fehlerhafte Werte übernommen werden.



Auch wenn man an den Ergebnissen sieht, dass sie ihren Zweck erfüllen, so bleiben sie dennoch unter den in dem Affektiven Nomen Datensatz enthaltenen Werten. Jedoch zeigen sie verschiedene Möglichkeiten leicht relativ sinnvolle Werte von einem kleineren Datensatz zu extrapolieren.

### 5.3 Anwendung auf den weiteren Datensätzen

Um unsere Ergebnisse genauer betrachten zu können verwenden wir die Features für die Klassifizierung für unsere anderen Datensätze, sowie zu guter Letzt die Verknüpfung aller drei. Genauer betrachten wir die jeweiligen besten Features in Tabelle 5.1, also Unigramme, deren Nomen durch ihren Cluster-Tag ersetzt wurden, den durchschnittlichen Abstraktheitswert der Nomen, den Distributionellen Fit von Basisverb im Vergleich zum Partikelverb, sowie die Kombination dieser 3 Features. Die Ergebnisse davon finden sich in Tabelle 5.3. Wie zu sehen ist erhalten wir bei der Kombination der Features im Allgemeinen dasselbe Ergebnis mit in etwa 80% Genauigkeit.

Die starke Tendenz der Sätze im Germanet Satz führt dazu, dass unser Klassifizierer versucht fast nur wörtlich auszugeben, was zwar zu einem besseren Ergebnis als in den anderen Datensätzen führt, dabei jedoch unter der Grenze der Baseline endet und die gewählten Features kaum verwendet.

Der dritte Datensatz lässt sich durch seine kurzen Sätze sehr gut mithilfe der Unigramme klassifizieren. Interessanterweise enthält dieser Datensatz das schlechteste Ergebnis für den Distributionellen Fit, obwohl er der einzige ist, der einige metaphorische Sätze richtig erkennt.

Die Summe der drei Datensätze folgt der Tendenz des NAACL Datensatzes, welcher mit knapp 70% der Datenmenge den deutlich größten Anteil der Sätze bereitstellt. Jedoch sehen wir hier eine leichte Verbesserung der meisten Features, insbesondere des F1-Maß für metaphorische Identifizierung, was an der größeren Datenmenge, aber auch einfach an der verschobenen Baseline liegen kann.

### 5.4 Duden-Sense

Für die Duden-Senses unterscheiden wir die zwei in Kapitel 4.3 beschriebenen Varianten. Die erste wird mithilfe eines Naive-Bayes Klassifizierers durch Vorkommen der jeweiligen Verb+Sense Verknüpfungen implementiert. Die Dimensionsgröße beläuft sich also auf die Menge der extrahierten Bedeutungen (die zusätzlich im Text erkannt werden müssen). Hierbei enthält die erste zum Teil das jeweilige Partikelverb. Bisher wurde versucht dies weitgehendst zu vermeiden, um die Bedeutung aus dem Kontext zu identifizieren. Auch wenn der Klassifizierer im ersten Moment nicht weiß, welches Partikelverb enthalten ist, so erhält er diese Information indirekt indem er für jedes Partikelverb die verschiedenen Bedeutungen lernt. Die zweite Implementierung geht, dies angehend einen Schritt weiter und übergibt die durchschnittlichen Abstraktheitswerte der ausgewählten Sense-Definition als Feature und vermeidet so unnötige Informationen. Die Verwendung des zuvor beschriebenen Lesk-Algorithmus resultiert hier zwar zu einigermaßen guten Ergebnissen, insbesondere die Verknüpfung von Partikelverb und Sense liefert eine

## 5 Experimente

Klassifizierung auf NAACL	Accuracy	Precision	Lit F1	Non-Lit F1
Baseline	0,65			
Unigramme (Nomen geclustert)	0,77 (+/- 0,04)	0,77	0,83	0,66
Abstraktheit von Nomen	0,74 (+/- 0,04)	0,73	0,82	0,55
Distr-Fit.	0,72 (+/- 0,04)	0,47	0,78	0,01
Kombination	0,80 (+/- 0,03)	0,79	0,84	0,69
Klassifizierung auf Germanet	Accuracy	Precision	Lit F1	Non-Lit F1
Baseline	0,86			
Unigramme (Nomen geclustert)	0,84 (+/- 0,07)	0,79	0,91	0,14
Abstraktheit von Nomen	0,86 (+/- 0,04)	0,74	0,92	0,00
Distr-Fit.	0,86 (+/- 0,03)	0,73	0,92	0,00
Kombination	0,82 (+/- 0,06)	0,79	0,90	0,20
Klassifizierung auf Spatial	Accuracy	Precision	Lit F1	Non-Lit F1
Baseline	0,65			
Unigramme (Nomen geclustert)	0,81 (+/- 0,08)	0,81	0,86	0,70
Abstraktheit von Nomen	0,75 (+/- 0,07)	0,74	0,83	0,52
Distr-Fit.	0,68 (+/- 0,07)	0,73	0,79	0,32
Kombination	0,80 (+/- 0,04)	0,79	0,86	0,69
Klassifizierung auf Allen	Accuracy	Precision	Lit F1	Non-Lit F1
Baseline	0,69			
Unigramme (Nomen geclustert)	0,78 (+/- 0,02)	0,77	0,84	0,61
Abstraktheit von Nomen	0,75 (+/- 0,03)	0,73	0,83	0,47
Distr-Fit.	0,73 (+/- 0,03)	0,72	0,83	0,39
Kombination	0,80 (+/- 0,03)	0,78	0,85	0,63

**Tabelle 5.3:** Ergebnisse der interessanten Features auf den verschiedenen Datensätzen, Kombination enthält alle 3 darüber genannten Features, Alle ist die Summe der 3 Datensätze

Verbesserung von 25% gegenüber der Baseline, dies sind bei einer Überdeckung von gerade einmal 106 Sätzen und einer Varianz von bis zu 28% jedoch kaum aussagekräftiges Ergebnisse.

Feature	Accuracy	Precision	Lit F1	Non-Lit F1
Baseline	0,56 (59:47)			
Partikelverb_sense	0,81 (+/- 0,28)	0,80	0,84	0,77
Duden durchschnittl. Abstraktheit	0,73 (+/- 0,20)	0,73	0,77	0,67

Die Überdeckung stammt aus der Menge der Sätze für die der Lesk-Algorithmus entscheiden kann, welche Bedeutung des Partikelverbs dazu gehört. Falls in den Definitionen des Partikelverbs keines der im Satz auffindbaren Wörter enthalten ist, so findet der Algorithmus keine Überdeckung. Dadurch wird die bereits geringe Menge der verwendbaren Sätze des Spatial-Datensatz (etwa 1200) stark gekürzt. Die niedrige Anzahl spiegelt sich in einer sehr hohen Varianz, also unzuverlässigen Ergebnissen, wieder.

Alternativ verwenden wir für die Identifizierung des Sense die Ähnlichkeit des Partikelverb Satzes zu den Verschiedenen im Duden aufgezählten Sätzen. Die durchschnittliche Ähnlichkeit der Wortvektoren von Kontextworten zu den im Duden stehenden bestimmt den Grad der Ähnlichkeit.

Feature	Accuracy	Precision	Lit F1	Non-Lit F1
Baseline	0,63 (439 : 261)			
1 Partikelverb_sense	0,83 (+/- 0,07)	0,83	0,88	0,77
2 Duden durschnittl. Abstraktheit	0,74 (+/- 0,11)	0,74	0,81	0,59
3 Unigramme (Nomen geclustert)	0,77 (+/- 0,08)	0,77	0,82	0,67
4 Abstraktheit von Nomen	0,73 (+/- 0,08)	0,73	0,79	0,61
5 Kombination	0,80 (+/- 0,07)	0,80	0,85	0,73
6 Kombination + 1	0,80 (+/- 0,07)	0,80	0,85	0,73
7 Kombination + 2	0,85 (+/- 0,10)	0,85	0,88	0,80

**Tabelle 5.4:** Ergebnisse der betrachteten Sense Varianten

Wie man an den Ergebnissen in Tabelle 5.4 sieht ähneln sich die Differenzen in Bezug auf die Baseline zu der des Lesk-Algorithmus. Das erste Feature erreicht separat sogar bessere Werte als unsere vorigen Ergebnisse, während das zweite mit den Abstraktheitswerten vergleichbare Ergebnisse liefert. Besonders interessant ist, dass obwohl das erste Feature seperat deutlich besser abschneidet, es in der Kombination in Overfitting Probleme gerät, während das zweite eine deutliche Verbesserung bietet. Hier erreichen wir eine Genauigkeit von 88%. Dennoch haben alle Lösung eine relativ starke Varianz und schwanken um bis zu 11%. Ein Grund dafür ist wieder die geringe Datenmenge. Dadurch das nur Teile eines Datensatzes verwendet werden können, beläuft sich diese auf 439 wörtliche und 261 metaphorische Sätze, insgesamt also gerade einmal 700 Sätze.



## 6 Qualitative Analyse

In diesem Kapitel wollen wir die zuvor ermittelten Ergebnisse genauer analysieren. Dafür stellen wir die Erfolgsraten der Klassifizierung in Relation zu den annotierten metaphorischen Werten und werfen einen genaueren Blick auf die Effekte der verschiedenen Partikel. Des Weiteren versuchen wir die Relation der Abstraktheit des Kontextes zu metaphorischer Sprache aufzeigen.

### 6.1 Annotierte Werte

Zunächst betrachten wir jeweiligen Resultate zu den von Menschen annotierten Werte. Hierbei werden die Daten über alle (bzw. auf den jeweiligen Kreuzvalidierungs-Splits) Daten trainiert, die Ergebnisse jedoch in die 5 Mengen bezüglich der metaphorischen Werte unterteilt. Aufgrund beschränkter Daten, werden nur der NAACL und der Spatial-Datensatz verwendet. In der Tabelle 6.1 können wir einige Dinge beobachten. Einerseits sehen wir an der Anzahl dazugehöriger Sätze, dass in unseren Datensätzen allen annotierten Werten der richtige binäre Wert für wörtlich/metaphorisch hinzugefügt wurde. Die hier gewählte Unterscheidung führt dazu, dass die Genauigkeit äquivalent zu der Trefferquote ist, dadurch, dass es in keinem Fall richtig Negative(TP) und falsch Positive geben kann (jeweils umgekehrt für [0-2,5) und [2,5-5]).

Interessanter wird es, wenn wir uns die eigentlichen Erfolgsraten anschauen. So sehen wir einerseits, dass in unserer Sprache meistens stark wörtliche oder stark metaphorische Verwendung findet, nur wenige Worte finden sich zwischen diesen Kategorien. Des Weiteren fällt es dem Klassifizierer, wie zu erwarten ist, deutlich schwerer Werte zu kategorisieren, welche nicht in die äußeren Kategorien fallen, was teilweise aber auch an der niedrigeren Menge an Trainingsdaten liegen kann. Ebenfalls scheinen wörtliche Verben besser identifiziert zu werden, so ist der Unterschied zwischen wörtlich und metaphorisch größer als der Bias zu wörtlichen Werten vermuten lässt.

N-Lit Wert	Accuracy	Precision	Lit F1	N-Lit F1	#Lit	#N-Lit
[0 – 1)	0,88	0,88	0,93	0,00	4336	0
[1 – 2)	0,77	0,77	0,87	0,00	459	0
[2 – 3)	0,76	0,76	0,87	0,00	87	0
[3 – 4)	0,48	0,48	0,00	0,65	0	277
[4 – 5]	0,71	0,71	0,00	0,83	0	2366

**Tabelle 6.1:** Ergebnisse zu annotierten N-Lit Werten, Klammern stellen Schranken dar, wobei eckige inklusive und runde ohne den gegebenen Wert sind

Partikel	Accuracy	Precision	Lit F1	N-Lit F1	#Lit	#N-Lit
bei	0,87	0,87	0,92	0,50	13	2
nach	0,80	0,79	0,87	0,46	252	61
ab	0,80	0,81	0,87	0,57	1304	379
ein	0,89	0,89	0,92	0,78	480	179
aus	0,78	0,79	0,84	0,62	1076	420
an	0,76	0,75	0,83	0,57	1290	607
zu	0,86	0,86	0,9	0,78	230	116
vor	0,77	0,78	0,81	0,70	53	29
auf	0,81	0,81	0,86	0,71	1553	902
mit	0,66	0,66	0,72	0,58	119	85
durch	0,67	0,93	0,33	0,78	11	121

Tabelle 6.2: Ergebnisse pro Partikel

## 6.2 Partikel

Da Partikelverben ein Fokus dieser Arbeit sind, wollen wir in diesem Abschnitt die dazugehörigen Partikel betrachten. Für die Auswertung auf dem Trainingsdatensatz, bestehend aus allen 3 Datensätzen, wird die 8 aus Tabelle 5.1 verwendet. Die dazugehörige Verteilung der Partikel findet sich in Abbildung 6.1. Dabei finden wir alle Partikel, mit Ausnahme von *durch* deutlich öfter in

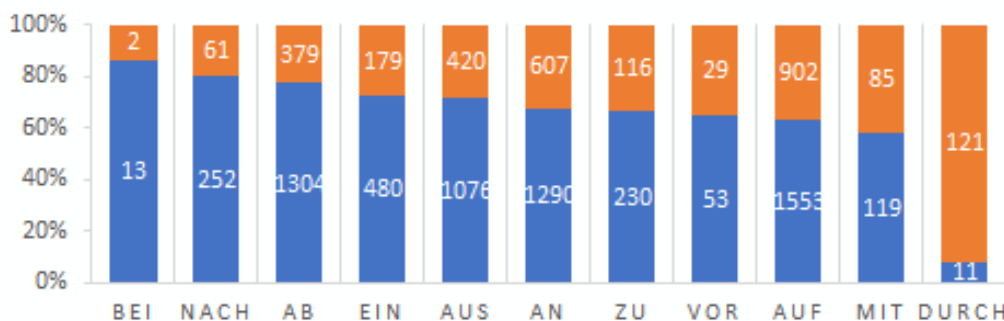


Abbildung 6.1: Partikel-Verteilung

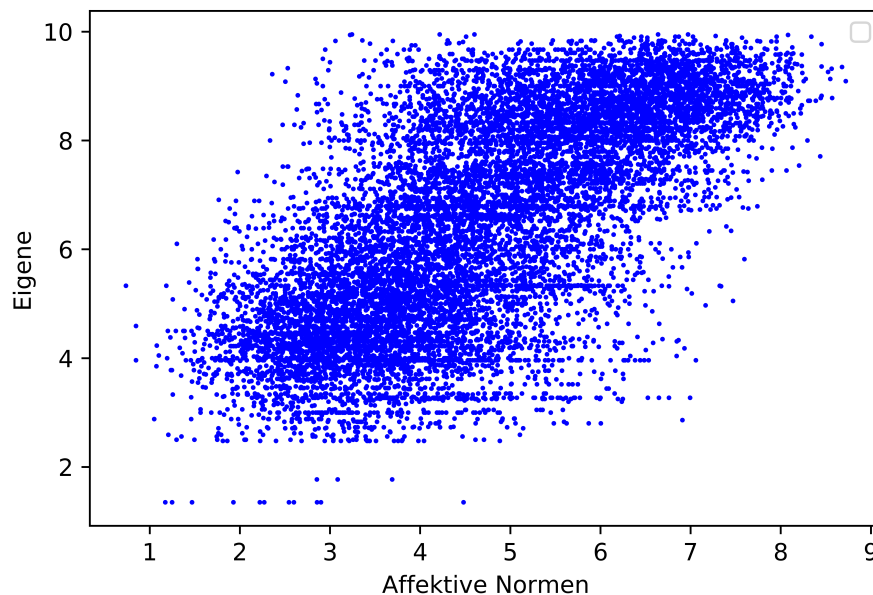
einer wörtlichen Verwendung. Zum Großteil liegt dies an der Baseline der ausgewählten Sätze von 68,7% zu den wörtlichen Sätzen, wenn man diese also mit einbezieht, liegt bei, nach, ab, ein, aus, an über, und zu, vor, auf, mit, durch unter dem Durchschnitt. Insbesondere fallen hier die Partikel *bei* und *durch* mit jeweils einem Bias von fast 90% auf. In beiden Fällen, besonders für *bei*, ist die Menge an vertretenen Sätzen jedoch vergleichsweise gering.

In Tabelle 6.2 finden wir die Ergebnisse zu den Partikeln. Wie zuvor sehen wir den Trend, dass metaphorische Sprache schwererer zu identifizieren ist. Besonders gute Ergebnisse erhalten wir für die Partikel *bei*, *ein* und *zu*. Dabei haben letztere weder besonders wenige, noch sonderlich viele Daten. Das Partikel *durch* erreicht eine überraschend schlechte Genauigkeit, trotz relativ hoher

Präzision. So sind zwar die meisten dem metaphorischen zugeordneten richtig zugeordnet, jedoch werden relativ viele fehlerhaft als wörtlich identifiziert. Ebenfalls zu beobachten ist, dass die am häufigsten vorkommenden Partikel von der Genauigkeit alle ziemlich mittig liegen.

### 6.3 Abstraktheitswerte

Zu guter Letzt wollen wir noch einen Blick auf unsere Abstraktheitswerte liefern. In der Abbildung 6.2 sehen wir für alle unsere Worte die dafür definierten Abstraktheitswerte. Auf der x-Achse sehen wir die von den affektiven Nomen gegebenen Werte, während auf der y-Achse unsere aus nur einem Wert bestehenden Abstraktheitswerte abbilden. Diese gehörten zwar zu den schlechteren Varianten, ergaben aber dennoch sinnvolle Ergebnisse. Im Graphen können wir leicht eine Korrelation der



**Abbildung 6.2:** Vergleich der eigenen Abstraktheitswerte mit denen in Affektive Normen

jeweiligen Abstraktheitswerte feststellen. Die Berechnung der Spearman-Korrelation ergab den Wert 0,69.





## 7 Zusammenfassung und Ausblick

In dieser Arbeit haben wir damit angefangen verschiedene Features auf dem NAACL Datensatz zu implementieren. Dabei betrachteten zunächst verschiedene Varianten von Unigrammen, Distributionellen Fit sowie Abstraktheit, um auf den besten Featuremengen weiter arbeiten zu können. Diese Feature Mengen erweiterten wir auf zwei weitere Datensätze und erreichten auf einer Menge von knapp 12.000 Sätzen eine Genauigkeit von 80%, eine 15% Verbesserung gegenüber der Baseline, mit einem F1-Maß von 0,85 für die Identifizierung von wörtlichen und 0,63 für die Identifizierung von metaphorischen Partikelverben.

Des Weiteren betrachteten wir verschiedene Alternativen, Abstraktheitswerte leicht aus einem kleineren Datensatz, mithilfe von Wortähnlichkeiten zu extrapolieren. Dabei stellten wir sowohl fest, dass der eigentliche Abstraktheitswert tatsächlich weniger wichtig ist als die Ähnlichkeit der Worte, als auch die Tatsache, dass das Erweitern der verglichenen Menge an Worten nicht unbedingt zu einem besseren Ergebnis führt. Ein weiteres betrachtetes Feature war die Verwendung der Wortbedeutung. Dabei unterschieden wir neben dem Lesk-Algorithmus wieder in zwei Varianten für die übergebenen Werte, um den Effekt des Partikelverbs zu reduzieren. Überraschender Weise sind wir in Kombination mit den anderen Features dazu in der Lage, mit einer der Methoden eine Genauigkeit von 85% mit einem F1-Maß von 0,88 für wörtliche und 0,80 für metaphorische Identifizierung. Besonders hebt sich hierbei die Erfolgsrate für metaphorische Partikelverben hervor, welche standardmäßig meist schlechte Werte vorwies. Hierbei sind wir jedoch durch eine geringe Datenmenge von gerade einmal 700 verwendeten annotierten Werten stark beschränkt und erhalten jeweils relativ hohe Varianzen.

Zu guter Letzt betrachteten wir unsere Ergebnisse nochmals genauer. Dabei identifizierten wir einerseits die Rolle der verschiedenen Partikel, betrachteten aber auch die Erfolgsraten für die Stufen [0,5] zur metaphorischen Sprache.

### Ausblick

Aus dieser Arbeit lassen sich verschiedene kleinere Themen entwickeln, welche man in Zukunft verfolgen kann. Allen voran die Bedeutung der Wort-Senses, wobei man die Menge an Daten erhöhen müsste, um relevante Ergebnisse zu erhalten.

Ebenfalls könnte man versuchen Abstraktheitswerte genauer durch eine Gewichtung der verschiedenen Referenzwerte zu bestimmen. Da die Auswahl aus konkreten und abstrakten Worten bestehen muss, besteht jedoch die Gefahr, dass die resultierenden Werte nur begrenzte Extreme enthalten, die Werte sich also alle ungefähr um den Median befinden.

Es lassen sich auch weitere, von dieser Arbeit unabhängige Features betrachten, welche interessante Eigenschaften repräsentieren können. Ein Beispiel, für das entsprechende Daten vorliegen, wäre die gefühlte Richtung von Menschen in Bezug zu einem Wort oder einer Phrase. Ansonsten ist es

auch immer möglich Forschungen zu neuronalen Netzen zu betreiben. Die Menge der vorhanden Arbeiten is vergleichsweise gering, während diese in manchen Bereichen der Computerlinguistik deutliche Verbesserungen erbrachten.

## Literaturverzeichnis

- [BBC+13] G. A. Broadwell, U. Boz, I. Cases, T. Strzalkowski, L. Feldman, S. Taylor, S. Shaikh, T. Liu, K. Cho, N. Webb. „Using Imageability and Topic Chaining to Locate Metaphors in Linguistic Corpora“. In: *Proceedings of the 6th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*. SBP'13. Washington, DC: Springer-Verlag, 2013, S. 102–110. ISBN: 978-3-642-37209-4. DOI: [10.1007/978-3-642-37210-0\\_12](https://doi.org/10.1007/978-3-642-37210-0_12). URL: [http://dx.doi.org/10.1007/978-3-642-37210-0\\_12](http://dx.doi.org/10.1007/978-3-642-37210-0_12) (zitiert auf S. 13).
- [BS06] J. Birke, A. Sarkar. „A clustering approach for nearly unsupervised recognition of nonliteral language“. In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. 2006 (zitiert auf S. 11, 13).
- [BS07] J. Birke, A. Sarkar. „Active learning for the identification of nonliteral language“. In: *Proceedings of the Workshop on Computational Approaches to Figurative Language*. Association for Computational Linguistics. 2007, S. 21–28 (zitiert auf S. 11, 13).
- [JM14] D. Jurafsky, J. H. Martin. *Speech and language processing*. Bd. 3. Pearson London: 2014 (zitiert auf S. 18).
- [KW16] M. Köper, S. S. im Walde. „Distinguishing literal and non-literal usage of German particle verbs“. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, S. 353–362 (zitiert auf S. 3, 11, 14, 15, 17).
- [Les86] M. Lesk. „Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone“. In: *Proceedings of the 5th annual international conference on Systems documentation*. ACM. 1986, S. 24–26 (zitiert auf S. 3, 21).
- [LJ80] G. Lakofi, M. Johnson. „Metaphors we live by“. In: *Chicago, IL: University of* (1980) (zitiert auf S. 13, 19).
- [Sch95] H. Schmid. „Improvements in part-of-speech tagging with an application to German“. In: *In proceedings of the acl sigdat-workshop*. Citeseer. 1995 (zitiert auf S. 15).
- [SL09] C. Sporleder, L. Li. „Unsupervised recognition of literal and non-literal use of idiomatic expressions“. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2009, S. 754–762 (zitiert auf S. 11, 13).
- [SS13] E. Shutova, L. Sun. „Unsupervised metaphor identification using hierarchical graph factorization clustering“. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, S. 978–988 (zitiert auf S. 11, 13).

- [TBG+14] Y. Tsvetkov, L. Boytsov, A. Gershman, E. Nyberg, C. Dyer. „Metaphor detection with cross-lingual model transfer“. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bd. 1. 2014, S. 248–258 (zitiert auf S. 14).
- [TMG13] Y. Tsvetkov, E. Mukomel, A. Gershman. „Cross-lingual metaphor detection using common semantic features“. In: *Proceedings of the First Workshop on Metaphor in NLP*. 2013, S. 45–51 (zitiert auf S. 14).
- [TNAC11] P.D. Turney, Y. Neuman, D. Assaf, Y. Cohen. „Literal and metaphorical sense identification through concrete and abstract context“. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2011, S. 680–690 (zitiert auf S. 11, 13, 19).

Alle URLs wurden zuletzt am 29.05.2018 geprüft.

### **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift