Zeitschrift: Annalas da la Societad Retorumantscha
Herausgeber: Societad Retorumantscha
Band: 136 (2023)

Artikel: Syntax und Semantik des Bündnerromanischen - Sprachsysteme und Sprachverarbeitung
Autor: Lutz, Florentin / Rolshoven, Jürgen
DOI: https://doi.org/10.5169/seals-1061910

Nutzungsbedingungen
Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Siehe Rechtliche Hinweise.

Conditions d'utilisation
L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. Voir Informations légales.

Terms of use
The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. See Legal notice.

Download PDF: 15.03.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Syntax und Semantik des Bündnerromanischen – Sprachsysteme und Sprachverarbeitung

Florentin Lutz & Jürgen Rolshoven *(Universität zu Köln)*

Abstract

La considerabla diversitad linguistica dils idioms dil romontsch ei ina sfida per saver descriver quei lungatg a moda holistica e per saver transponer quella descripziun en la linguistica computaziunala per la translaziun maschinala. La contribuziun suandonta applichescha metodas innovativas per tractar sintaxa e semantica dil romontsch e metta en evidenza lur muntada per descripziuns linguisticas. La contribuziun remplazza in model da cumpetenza static che vegn buca da declarar che ins sa capir in lungatg era incumplet e cun sbagls. Quei fa la metoda presenta cumbinond proceders sintactics e semantics che ein dinamics e che san divergir. Ultra da quei muossa quella contribuziun co resursas linguisticas san vegnir tratgas a nez.

sintaxa / syntax – semantica / semantics – semantica vectoriala / vector semantics – emprender automatic / machine learning – translaziun automatica / machine translation

1. Das Bündnerromanische in systemlinguistischer Sicht

1.1. Anmerkungen zur Linguistik des Bündnerromanischen

Sprachliche Systeme sind – wie das Bündnerromanische und seine Idiome eindrücklich zeigen – dynamisch. Eine Dynamik setzt ein System voraus, das in der Prozessierung ein gewisses Mass an (Fehler-)Toleranz besitzt. Das erfordert den Abschied von einem hochgradig statischen Kompetenzmodell, das nur grammatisch vollständige Strukturen erkennt und verarbeitet. Gefordert ist hingegen die Fähigkeit, Ungrammatikalität zu erkennen und gleichwohl zu prozessieren. Akzeptierte und prozessierbare Fehlertoleranz ist eine Voraussetzung für die Möglichkeit der Restrukturierung des sprachlichen Systems und seiner Evolution.

Dies aber ist nur ein Aspekt eines toleranten Systems, eines Systems, das unvollständigen Input zu analysieren vermag. Ein anderer Aspekt ist der eines nicht vollständigen Systems, das gleichwohl aufgrund seiner Toleranz sprachlichen Input zu verarbeiten vermag, der aufgrund begrenzten linguistischen Wissens nur partiell strukturell beherrscht wird. Das befähigt ein maschinelles System dazu, auch solche sprachlichen Äusserungen zu verarbeiten, die sein linguistisches Wissen überschreiten. Damit ähnelt es einem Sprecher mit begrenzter Fremdsprachenkompetenz, der trotz fremdsprachlicher Unzulänglichkeit in der Lage ist, Äusserungen zu verstehen.

Ein pragmatisches Vorgehen ist für Kleinsprachen (und deren regionale Varietäten) auch insofern geboten, als die Zahl von Spezialisten, die

vollständige syntaktisch-semantische Beschreibungen liefern könnten, eher klein ist.

Der vorliegende Beitrag fokussiert auf die fertiggestellten Module eines sprachverarbeitenden Systems; das System ist im Prinzip lauffähig, bedarf aber zu einer praktisch brauchbaren Nutzung einiger Ergänzungen, ist also *work in progress*.

1.2 Ein hybrider holistischer Ansatz

Der vorliegende Ansatz ist holistisch: Dank dynamischer Verfahren wird nicht nur unvollständiger Input beherrscht, sondern auch der Umgang mit unvollständigem linguistischem Wissen. Der Ansatz ist hybrid, insofern er Syntax und Semantik nicht starr vollsymbolisch, sondern durch spezielle Algorithmen und Datenstrukturen ähnlichkeitsbasiert prozessiert.

1.3 Sprachpraktische Folgerungen

Als ein Seiteneffekt eines validen linguistischen Ansatzes ergibt sich die maschinelle Übersetzung. Allerdings unterscheiden sich die hier vorgeschlagenen Verfahren von pragmatisch erfolgreichen maschinellen Übersetzungen wie *DeepL, Google Translator, Machine Translate of European Language Technology* oder für das Bündnerromanische (Rumantsch Grischun) *Textshuttle* der Zürcher Gruppe um Martin Volk. Im Gegensatz zu den erwähnten Systemen sind wir in der Lage, linguistisches Wissen explizit zu formulieren und somit im Sinne von Karl Popper falsifizierbar zu halten. Unsere Verfahren erzeugen vollsymbolische Repräsentationen, wohingegen das linguistische Wissen z. B. bei *DeepL* und *Textshuttle* subsymbolisch in den neuronalen Netzen verborgen bleibt.

2. Dynamiken des Sprachsystems

2.1 Semantik

2.1.1 Probleme traditioneller Semantik

Traditionelle Semantiken sind in formaler Sicht unbefriedigend, da sie ein sprachliches Vorverständnis voraussetzen. Merkmalssemantiken verwenden sprachliche Merkmale, z. B. ‘menschlich’ für ‘Kind’, Prototypensemantiken verwenden Prototypen, die sprachlich benannt sind (z. B. ‘Spatz’). Beide funktionieren für menschliche Nutzer, nicht aber formal oder maschinell. Letzteres zeigen die langen Misserfolge maschineller Übersetzung, die an Problemen lexikalischer Semantik scheiterten.

Ein weiterer Mangel der genannten Semantiken liegt daran, dass sie händisch erstellt werden müssen. Es erwies sich als praktisch nicht möglich, auf diese Weise konsistente umfangreiche Semantiken zu erstellen; Ähnliches gilt für relational orientierte Wortsemantiken.

Die aufgeworfene Problematik lässt sich nur dann lösen, wenn die semantische Beschreibungssprache auf ein anderes als ein sprachliches semiotisches System zurückgreift. Auf ein anderes beschreibendes System zurückzugreifen ist ein in den Naturwissenschaften übliches Verfahren. In der Chemie wird Kohle nicht durch Kohle (oder durch Nicht-Eisen) beschrieben, sondern durch ein differentes, nicht materielles semiotisches System, das wesentliche Eigenschaften von Kohle widerspiegelt. Sicherlich ist diese Situation nur begrenzt mit der hier aufgeworfenen semantischen Problematik vergleichbar; in der Chemie wird ein materielles System durch ein semiotisches beschrieben, während in der Semantik ein semiotisches System durch ein (anderes) semiotisches System zu beschreiben ist. Es ist festzuhalten, dass die beiden semiotischen Systeme different sein müssen.

2.1.2 Vektorsemantiken

Dazu werden hier nun zwei Verfahren vorgestellt. Das erste wurde bereits Ende der 1960er-Jahre von Gerard Salton vorgeschlagen (cf. Salton & McGill 1987), das zweite geht auf Mikolov et al. (2013) zurück.

Es handelt sich um Vektorsemantiken, die auf Gebrauch und Distribution in Textkorpora fussen.[1] Dieser Ansatz erfüllt nicht nur die Forderung nach einem differenten semiotischen Beschreibungssystem. Er ermöglicht es auch, Semantiken maschinell zu erzeugen.

Aus dem Vergleich semantischer Vektoren lassen sich semantische Ähnlichkeit und semantische Unschärfe berechnen – als Cosinusähnlichkeit. Dabei ergeben sich Werte zwischen 1 (Identität) und 0 (maximale Unähnlichkeit).

Einführend wird das Modell von Salton & McGill (1987) in stark vereinfachter Form anhand des folgenden artifiziellen Korpus vorgestellt: *Maria conta ina canzun. Maria legia in cudisch. Maria legia ina gasetta. Giusep legia in cudisch.*

Daraus wird eine Matrix erstellt, in deren erster Zeile und erster Spalte die Worttypen des Korpus alphabetisch sortiert eingetragen werden. In die Matrix wird nun eingetragen, wie oft unterschiedliche Wörter im gleichen Kontext – hier im gleichen Satz(kontext) vorkommen.

1 Einführend zu Vektoren z. B. Schirothek & Scholz 2005: 51.

Maria und *legia* kommen z. B. gemeinsam im zweiten und dritten Satz vor, insgesamt also zweimal.

	canzun	*conta*	*cudisch*	*gasetta*	*Giusep*	*in*	*ina*	*legia*	*Maria*
canzun	0	1	0	0	0	0	1	0	1
conta	1	0	0	0	0	0	1	0	1
cudisch	0	0	0	0	1	2	0	2	1
gasetta	0	0	0	0	0	0	1	1	1
Giusep	0	0	1	0	0	1	0	1	0
in	0	0	2	0	1	0	0	2	1
ina	1	1	0	1	0	0	0	1	2
legia	0	0	2	1	1	2	1	0	2
Maria	1	1	1	1	0	1	2	2	0

Tabelle 1: Matrix des Korpus

Daraus ergibt sich für z. B. *conta* der Vektor 1 0 0 0 0 0 1 0 1, für *legia* der Vektor 0 0 2 1 1 2 1 0 2.

Mit dem Ansatz von Salton sind zwei Probleme verbunden. Wegen der alphabetischen Sortierung der Worttypen ist der Ansatz sprachabhängig bzw. präziser *signifiant*-abhängig; für andere Sprachen ergeben sich nämlich auch bei gleichen Inhalten wegen anderer Worttypen (~ *Signifiants*, z. B. deutsch *Baum* vs. surselvisch *plonta*) andere Sortierungen und somit andere Vektoren. Ferner enthalten grosse Korpora sehr viele Worttypen, und da die meisten nicht in gleichen Kontexten auftreten, enthalten die Matrizes viele Nullen. Die Vektoren werden auf diese Weise wenig prägnant. Dieses Problem wird in der Literatur als *sparse data problem* bezeichnet.

Diese beiden Probleme wurden durch das Word2Vec-Verfahren von Mikolov et al. (2013) gelöst. Word2Vec begrenzt die Zahl der Komponenten eines Vektors (seine Länge) auf 100–1000 (oftmals 300). Der grösste Vorteil von Word2Vec (im Vergleich zu dem Verfahren von Salton) liegt in der *Signifié*-Orientierung der Vektoren, d. h. unterschiedliche *Signifiants* (aus unterschiedlichen Sprachen) wie oben deutsch *Baum* und surselvisch *plonta* haben gleiche (oder zumindest sehr ähnliche) Vektoren, welche ein gemeinsames *Signifié* ausdrücken. Darüber hinaus lassen sich grundlegende semantische Beziehungen aus dem Cosinusvergleich der Vektoren ableiten, z. B. paradigmatische semantische Beziehungen, Hyponymie, Hyperonymie oder Meronymie. Zusammengefasst: Aus Korpora auch unterschiedlicher Sprachen werden für gleiche oder ähnliche Bedeutungen ähnliche Vektoren erzeugt. Damit wird Word2Vec zu einem wertvollen

Werkzeug für formale Semantiken und maschinelle Sprachverarbeitung. Die rezenten Erfolge maschineller Übersetzung (z. B. *DeepL, TextShuttle*), aber auch die semantische Suche von Suchmaschinen *(Google)* beruhen auf Vektorsemantiken.

Die vorliegende Untersuchung fokussiert im Folgenden die Bereiche lexikalische Semantik[2] und Polysemie, Valenzrahmen, thematische Rollen und Subkategorisierung und leitet zu syntaktischen Fragestellungen über.

Aus der Analyse der Formen semantischer Ähnlichkeit ergibt sich die Möglichkeit der Aufdeckung von Polysemie. In den Kontexten des surselvischen Verbs *magliar* 'essen' / 'fressen' finden sich Formen wie *um, vacca, affon, femna, nuorsa, tgaun*. Ordnet man diese Kontextformen in einem Clusterverfahren nach semantischer Ähnlichkeit, so ergeben sich zwei Cluster: *um, femna, affon* und *vacca, nuorsa, tgaun*.

Das Stichwort Kontextanalyse führt zu lexikalisch festgelegten Kontexten, zu Subkategorisierungen, Valenzrahmen oder auch thematischen Rollen. Anstelle händisch gesetzter Merkmale (z. B. 'menschlich') wird hier der Vektor für einen prototypischen Repräsentanten gesetzt, im Falle von 'menschlich' für das Surselvische der Vektor für *femna* oder *um*. Bei der Auswahl von Kontexten in einer syntaktischen Sprachverarbeitung ist dann der Kontext mit der geringsten semantischen Differenz auszusuchen. Falls es nur grosse Differenzen gibt, könnte es sich um einen metaphorischen Kontext handeln – und somit das Problem der maschinellen Verarbeitung von Metaphern lösen.

Semantische Vektoren können nach dem Word2Vec-Verfahren auch mit Hilfe von Open Source Software aus Textkorpora erzeugt werden oder vorhandenen Vektorsammlungen (cf. Grave et al. 2018) entnommen werden. Für das Bündnerromanische und seine Idiome sind beide Vorgehensweisen kritisch zu betrachten, da zum einen hinreichend grosse Textkorpora zur Erfassung eines grösseren Vokabulars fehlen, zum anderen die vorhandene Vektorsammlung auf der vergleichsweise kleinen Wikipedia (in Rumantsch Grischun) beruht.

2 Hier sei kurz auf weitere Potentiale vektorsemantischer Methoden verwiesen. Aus einer vollständigen Auswertung eines etymologischen Wörterbuchs lassen sich durch den vektoriellen Vergleich der Semantik der Etyma und der resultierenden Wortformen diachrone und diatopische semantische Dynamiken holistisch aufdecken. Damit ist ein etymologisches Wörterbuch nicht lediglich die Summe seiner Etyma, sondern ein Instrument zur Aufdeckung von semantischen Dynamiken im Sprachsystem.

Allerdings verfügen das Bündnerromanische und seine Idiome über eine grosse, zum Teil auch digitalisierte[3] Sammlung von Wörterbüchern, in denen die Bedeutung der Lemmata durch Äquivalente (meist in Deutsch) aufgeführt sind. Den Bedeutungen der Äquivalente werden aus der oben genannten aufgeführten Vektorsammlung des Deutschen die entsprechenden Vektoren entnommen. Werden für eine Lesart eines bündnerromanischen Lemmas mehrere Bedeutungsnuancen im Deutschen angegeben, so wird der bündnerromanische Vektor durch Mittelwertbildung der Vektoren der deutschen Bedeutungsnuancen rekonstruiert.

Die aus dem Word2Vec-Verfahren folgende *Signifié*-Orientierung hat einen weiteren, unseres Erachtens bemerkenswerten Seiteneffekt: Sie macht zweisprachige Wörterbücher überflüssig. Sie ermöglicht es, zwei einsprachige Wörterbücher über die ähnlichsten semantischen Vektoren dynamisch zu verknüpfen. Damit wird der lexikographische Aufwand, der sich für die Erstellung von zweisprachigen Wörterbüchern gerade für Kleinsprachen mit einer grösseren Zahl regionaler Varietäten ergibt, stark reduziert. Weitere Sprachpaare – etwa surselvisch-spanisch oder engadinisch-katalanisch, surmeirisch-englisch – lassen sich mit geringem Aufwand hinzufügen.

2.2 Syntax

Ein distributionelles Verfahren wie Word2Vec ist seiner Natur nach von Kontexten abhängig. Dies gilt für die Erstellung von Vektoren, aber auch für deren Nutzung, wie oben für Valenzrahmen und Subkategorisierung angedeutet wurde. Semantik und Syntax überschneiden sich. Dabei stellt sich die Frage, ob die durch Vektorsemantiken gewonnene Flexibilität und Dynamik auf Syntax und syntaktische Verarbeitung übertragen werden kann, nicht zuletzt auch vor dem Hintergrund der einleitend aufgeworfenen Fragen zur Statik und Inflexibilität eines zu strikten Kompetenzmodells.

Aussagen werden in einem solchen Modell oftmals in Gestalt von wenn-dann-Beziehungen gemacht, z. B. in der Form:

> Wenn eine Kategorie A ein Merkmal x und eine Kategorie B ein Merkmal y trägt, dann führe eine Aktion Z durch.

3 Liegt ein Wörterbuch als PDF vor (z. B. Decurtins 2001 oder Decurtins 2012), so kann mit Hilfe von Open source Software daraus ein XML-Dokument gewonnen werden. Ein solches Dokument spiegelt das Layout des Wörterbuchs in Gestalt einer Baumstruktur wider. Daraus werden die Makro- und Mikrostruktur leicht zugreifbar.

Formal entspricht der Wenn-Teil einer Konjunktion von Prädikaten, die zur Erfüllung der Aktion alle nach (logisch) wahr evaluiert werden müssen. Ist das nicht der Fall, so misslingt der Aufbau einer Struktur.

Letztlich wird auf diese Weise Sprache als eine Menge von Sätzen verstanden, die nur streng regelbasiert erzeugt oder verstanden werden kann. Im Sprachgebrauch zeigt sich jedoch, dass auch in dieser Sicht ungrammatische Sätze verstanden werden. Die aufgeworfene Problematik kann nicht darüber abgetan werden, den Sprachgebrauch auszuklammern. Vielmehr bedarf es einer abweichungstoleranten Sprachverwendung; ohne sie ist Sprachevolution nicht erklärbar – wäre dem nicht so, wäre noch heute Latein die Umgangssprache Romanischbündens.

Im Folgenden wird ein abweichungstoleranter Ansatz auch für die syntaktische Verarbeitung des Bündnerromanischen skizziert; die Grundidee einer dynamischen und ähnlichkeitsorientierten Semantik wird für die Syntax und ihre Verarbeitung übertragen.

Eine schrittweise, quasi atomare Prozessierung wird dabei durch eine holistische Parallelverarbeitung substituiert. Dies kommt auch Ideen massiv-paralleler Verarbeitung im menschlichen kognitiven Apparat näher.

In unterschiedlichen syntaktischen Theorien werden Strukturbäume oder Stemmata verwendet, um Hierarchie und Reihenfolge auszudrücken. Für eine flexible und abweichungs- oder fehlertolerante Prozessierung ist es notwendig, Strukturen oder Teilstrukturen auch nach Ähnlichkeit zu vergleichen.

Baumgraphen sind allerdings zweidimensional, und zweidimensionale Strukturen sind schwierig zu vergleichen. Hingegen sind eindimensionale Strukturen – z. B. Zeichenketten – leichter zu vergleichen. Daher liegt es nahe, Baumstrukturen in Zeichenketten zu verwandeln, d. h. in Klammerstrukturen (cf. unten Abb. 1).

Für den Vergleich von Zeichenketten gibt es drei wesentliche Algorithmen, die mit unterschiedlichen Zielsetzungen verbunden sind. Der Levenshtein-Algorithmus eignet sich am besten für einen globalen Vergleich; er notiert die Zahl der Abweichungen in den beiden verglichenen Zeichenketten. Der Needleman-Wunsch-Algorithmus notiert global die Übereinstimmungen von zwei Zeichenketten, der Smith-Waterman-Algorithmus hingegen die lokalen Übereinstimmungen (cf. Gusfield 1997: 216).

Für die Prozessierung von Sprache – sei es die Analyse oder die Produktion eines Satzes – ist der Vergleich von gegebenen Teilstrukturen mit vorgegebenen Mustern hilfreich (cf. unten Abb. 1 und 2). Es ist festzuhalten, dass die Muster keinesfalls vollständig sein müssen; vielmehr

reicht es, im Falle von alternativen Mustern (z. B. für Haupt- oder für Nebensatz) den niedrigsten Abweichungswert für die gewünschte Struktur zu erzielen. Der vorliegende Ansatz operiert sowohl mit unvollständigen (auch im Aufbau befindlichen) Eingaben als auch mit unvollständigen Mustern.

Bei der analytischen Prozessierung von Sätzen sind Einträge im Lexikon mit aufzubauenden Strukturen abzugleichen. Das dient u. a. dazu, bei einem Verb beispielsweise die richtige Lesart im Lexikon zu finden und dann für den Aufbau von Strukturen zu nutzen. Auch dafür bedarf es des Vergleichs von Strukturbäumen. Es gibt den oder die (rudimentären) Strukturbäume für die Lesarten des Verbs in einem Lexikon, die mit dem zu verarbeitenden Satz verglichen («gematcht») werden (cf. unten Abb. 3, 4 und 5).

Abschliessend sei kurz darauf verwiesen, dass sich der Levenshtein-Algorithmus für die Erkennung komplementärer Strukturen wenig eignet, da er globale Ähnlichkeit erfasst.[4]

2.3 Syntax und Semantik

Es liegt nahe, die flexible und dynamische Prozessierung von Syntax und Semantik nicht mehr länger isoliert durchzuführen, sondern zu vereinen. Dies ist einfach umzusetzen. In den durch Zeichenketten realisierten Strukturbäumen werden die lexikalischen Einträge durch die zugehörigen Vektoren ersetzt. Beim Vergleich der Zeichenketten durch die oben genannten Algorithmen werden nun nicht nur Symbole – z. B. Kategorien und Klammern – auf Gleichheit oder Ungleichheit untersucht, sondern zusätzlich Vektoren. Für den Cosinusvergleich der Vektoren werden Schwellwerte definiert, bei deren Erreichen Gleichheit oder Ungleichheit gesetzt wird.

Der Vorteil dieses Vorgehens ergibt sich beispielsweise bei der Auswertung von Strukturen des Lexikons.

Enthält das Lexikon für das Lemma *schlachten* das Beispiel *der Bauer schlachtet die Kuh*, so werden die Einträge *Bauer* und *Kuh* durch ihre Vektoren ersetzt. Diese Vektoren sind aber den Vektoren von *Metzger* und *Schwein* im Beispielsatz *der Metzger schlachtet das Schwein* hinreichend ähnlich, quasi prototypisch. Das konkrete Vorkommen des Lexikons wird also zu einem prototypischen Strukturmuster.

4 Komplementär distribuierte Strukturen (z. B. surs. *Maria cumpra oz in niev cudisch* vs. *Oz cumpra Maria in niev cudisch*) lassen sich hingegen durch sukzessive Anwendung des Needleman-Wunsch-Algorithmus erkennen, der schrittweise die beste lokale Übereinstimmung liefert.

3. Ressourcen, Prozessierung und Potentiale

Es wurde einleitend darauf verwiesen, dass der vorliegende Ansatz *work in progress* ist. Er integriert eine grössere Anzahl bereits bestehender Module. Zu den Vorteilen modularer Softwarearchitekturen zählen die Möglichkeiten der Wiederverwendung von Bausteinen in anderen Kontexten. Dank der syntaktischen und semantischen Flexibilität dürfte es relativ einfach sein, Suchmaschinen für Korpora zu bauen, die prototypische syntaktische und semantische Muster enthalten. Dies dürfte die Produktivität von Hypothesenbildung, -verifikation oder -falsifikation steigern. Sicherlich ist zu berücksichtigen, dass die zum Einsatz kommenden Algorithmen in einem Informatiksinn komplex – d. h. mit einer sehr hohen Zahl von Rechenschritten verbunden – sind.

Dies ist aber insofern unproblematisch, als aufwändige Berechnungen dank Verlagerung auf Graphikkarten von PCs parallelisiert werden können. Damit werden Rechenkapazitäten zugänglich, die vor einem Jahrzehnt Grossrechner erforderten.

Abschliessend wird das praktische Potential des vorliegenden Ansatzes an dem Beispielsatz *Heute liest das Kind die Zeitung* und seiner Übersetzung schrittweise demonstriert. Im Prinzip besteht das Vorgehen aus zwei Schritten, der Analyse (*Parse*) des quellsprachlichen deutschen Satzes und der Erzeugung des entsprechenden zielsprachlichen surselvischen Satzes.

Für den deutschen quellsprachlichen Satz (nach Abgleich mit einem Lexikon des Deutschen in folgender Form: [ADV *Heute*] [V *liest*] [D *das*] [N *Kind*] [D *die*] [N *Zeitung*]) ist zu klären, ob es sich um einen Haupt- oder um einen Nebensatz handelt, d. h. ob das Verb an zweiter oder an finaler Position steht. Daher wird ein Abgleich eines Musters für einen Hauptsatz mit der (noch) unvollständigen Eingabe vorgenommen, als Beispiel für die oben beschriebene (abweichungs)tolerante syntaktische Prozessierung. Dabei zeigt sich, dass das Hauptsatzmuster am besten passt. Hier ist darauf zu verweisen, dass der vorgenommene Vergleich der syntaktischen Strukturen deren Umwandlung in Zeichenketten voraussetzt.

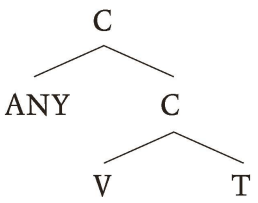

Abb. 1 : Muster als Baumstruktur

Als Zeichenkette ergibt sich daraus: [C[ANY] [C [V] [T]]][5].

Für das Vorkommen gilt: [ADV *Heute*] [V *liest*] ...

Der Zeichenkettenvergleichsalgorithmus (Levenshtein) ermittelt für die Hauptsatzverbstellung eine niedrigere Abweichung als für die Nebensatzstellung (hier nicht aufgeführt). Daher wird für das Verb die zweite Position angenommen (cf. unten die vollständige Struktur Abb. 6).

Levenshtein										
	[C	[ANY	]	[C	[V	]	[T	]	]	]
[ADV	1	1	2	3	4	5	6	7	8	9
Heute	2	2	2	3	4	5	6	7	8	9
]	3	3	2	3	4	5	5	6	7	8
[V	4	3	3	3	4	5	6	6	7	8
liest	5	4	4	4	4	5	6	7	7	8
]	6	5	4	5	5	5	5	6	7	7

Abb. 2: Strukturmuster und Strukturvorkommen im Zeichenkettenvergleich (Levenshtein-Algorithmus)

Die lexikalische Disambiguierung war lange eines der Hauptprobleme der maschinellen Übersetzung. Im Folgenden wird die Lösung dieses Problems durch Vektorsemantiken demonstriert.

Ein Verb wie *lesen* hat zwei Bedeutungen, wie *Bücher lesen* und *Weintrauben lesen* zeigen. Die Bedeutung von *lesen* hängt also von dem selegierten Objekt ab. Im Lexikon wird das selegierte Objekt durch dessen semantischen Vektor repräsentiert, im vorliegenden Fall durch den Vektor von *Buch*, unter Nutzung von dessen quasi prototypischer Ähnlichkeit zu den Vektoren von *Zeitung*, *Zeitschrift* u. a. m. Das Lexikon enthält als Muster einen Teilstrukturbaum, der die Beziehung des Verbs zu seinem Objekt widerspiegelt. Dies ermöglicht wiederum den Einsatz eines zeichenkettenvergleichenden Algorithmus, der zusätzlich vektorielle Ähnlichkeit (konkret: Cosinusähnlichkeit) berechnet. Im Folgenden zunächst das Muster im Lexikon und die Teilstruktur der syntaktischen Analyse:

5 ANY ist ein kategoriales Stellvertretersymbol, das für jede (vollständige, d. h. maximale) Kategorie steht. Es sei darauf verwiesen, dass bei einem flexiblen Abgleich von Strukturen durch Zeichenketten auf Angabe von Töchterknoten verzichtet werden kann.

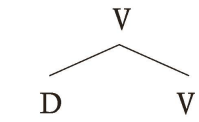

Abb. 3: Lexikoneintrag[6] als Baumstruktur

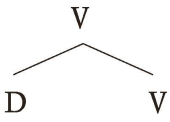

Abb. 4: Teilbaum der syntaktischen Analyse

Levenshtein								
	[V	[D	Vkt Buch	]	[V	lesen	]	]
[V	0	1	2	3	4	5	6	7
[D	1	0	1	2	3	4	5	6
VktZeitung	2	1	0	1	2	3	4	5
]	3	2	1	0	1	2	3	4
[V	4	3	2	1	0	1	2	3
lesen	5	4	3	2	1	0	1	2
]	6	5	4	3	2	1	0	1
]	7	6	5	4	3	2	1	0.12

Abb. 5: Zeichenkettenvergleich syntaktischer Strukturen

Nach Umwandlung der Strukturbäume in Zeichenketten wird die Lesart *Bücher lesen* selegiert, da die Vektordifferenz von *Zeitung* und *Buch* geringer ist als die des alternativen Eintrags von *lesen* mit dem Vektor für *Traube*[7].

6 'VktBuch' steht für den Vektor von Buch, einen Vektor mit 300 Komponenten.

7 Die Vektorähnlichkeit wird erst in dem äussersten unteren Kästchen der Matrix notiert; dies hängt damit zusammen, dass Vektorähnlichkeit nur für zwei Vektoren (cf. Vkt-Symbole), nicht aber zwischen einem Vkt-Symbol und einem anderen Symbol (Klammer oder Kategorie) notiert wird. Die Vektorähnlichkeit wird als Cosinus zweier Vektoren berechnet. Vektoren sind identisch oder sehr ähnlich, wenn der Cosinus-Wert 1 oder nahe 1 liegt, z. B. 0.9 oder 0.88 (im vorliegenden Fall). Da der Levenshtein-Algorithmus jedoch fehlerorientiert ist und Abweichungen durch eine 1, Gleichheit durch einen 0-Wert notiert, wird für den Cosinuswert für semantische Vektoren von 1 subtrahiert, so dass sich 0.12 als Wert ergibt, der starke Ähnlichkeit ausdrückt.

Als Ergebnis der Analyse ergibt sich insgesamt folgende Struktur:

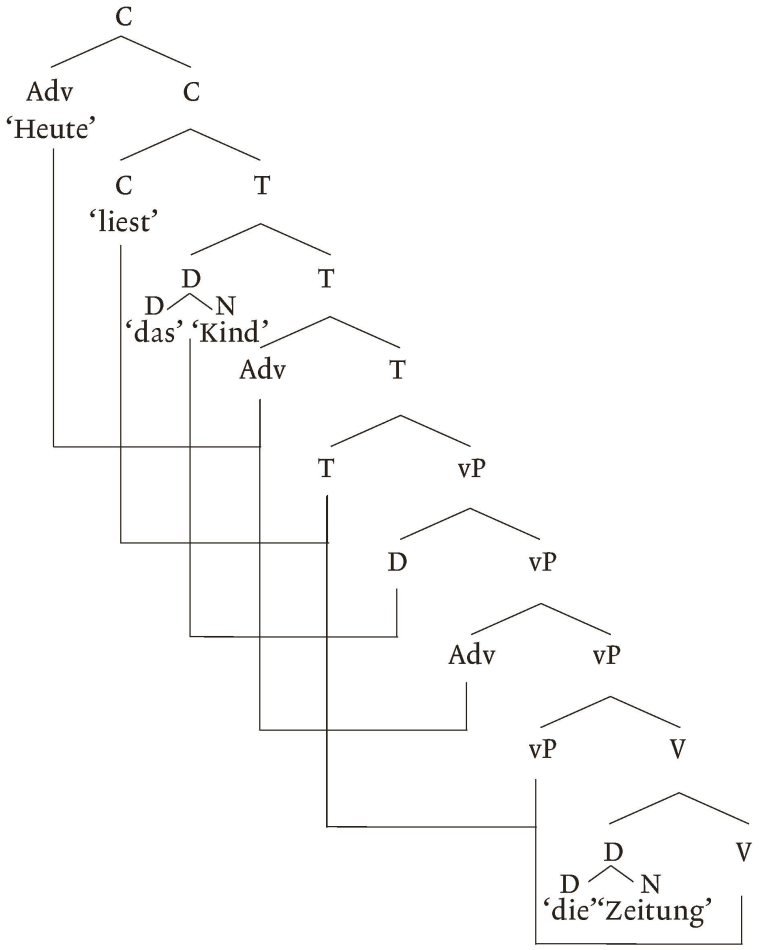

Abb. 6: Strukturbaum Quellsprache

Aus dem quellsprachlichen Baum wird nun der zielsprachliche Baum erzeugt; dabei wird der quellsprachliche Baum traversiert, um das Hauptverb zu finden (cf. Abb. 6, Kategorie V rechts unten, Ursprung der Bindungslinie).

Anschliessend werden – im Baum von unten nach oben vorgehend – die Mitspieler des Verbs, Aktanten und Zirkumstanten, gesucht (und gegebenenfalls wiederum deren Mitspieler). Für jede aufgefundene lexikalische Kategorie wird der semantische Vektor im einsprachigen quellsprachlichen Wörterbuch aufgesucht. Im (gleichfalls einsprachigen) zielsprachlichen Wörterbuch wird als lexikalische Entsprechung der Eintrag mit dem ähnlichsten Vektor ermittelt.[8]

8 Dank einer dynamischen Suche von Übersetzungsäquivalenten in einsprachigen Wörterbüchern entfällt die Notwendigkeit von Transferlexika, deren Zahl ja mit verschiedenen Quell- und Zielsprachen stark wächst. Die Vektoren für das zielsprachliche surselvische Wörterbuch können nicht aus digitalen Korpora gewonnen werden, da diese nicht die nötige Grösse besitzen. Daher werden sie aus den Vektoren der deutschen Äquivalente zweisprachiger Wörterbücher berechnet.

Mit den für die Erstellung der Wortfolge nötigen Bindungen entsteht folgende Baumstruktur, für die sich durch Auslesen der terminalen Knoten folgender zielsprachlicher Satz ergibt:

Oz legia igl affon la gasetta.

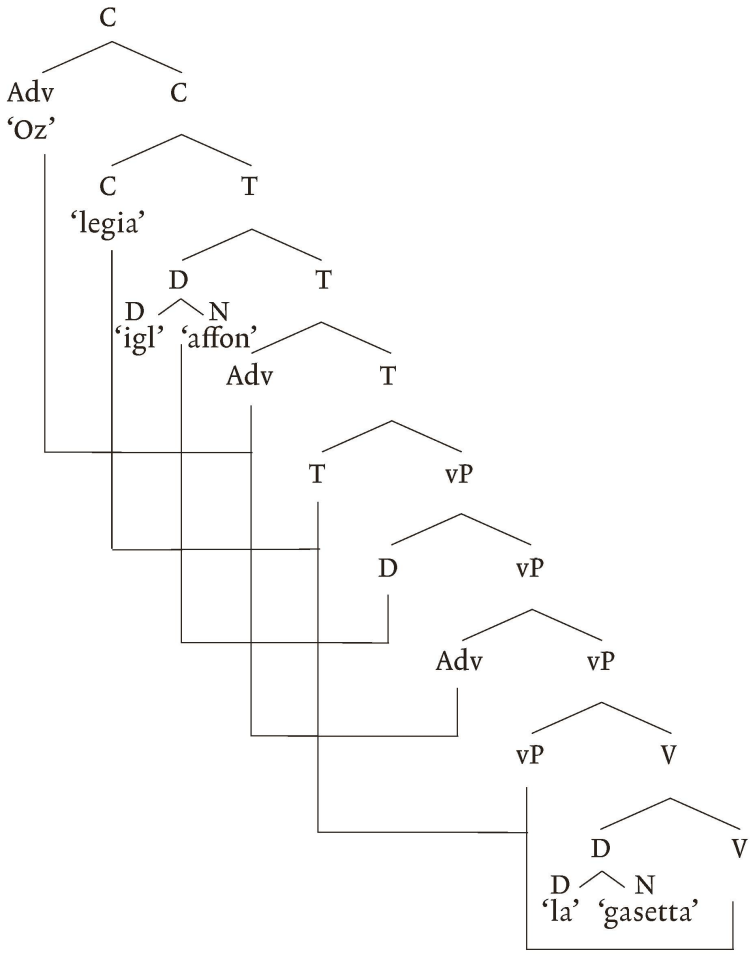

Abb. 7: Strukturbaum Zielsprache

Bibliografie

Anderson, Stephen (2006), *Verb Second, Subject Clitics, and Impersonals in Surmiran (Rumantsch)*, in: *Proceedings of the Annual Meeting of the Berkeley Linguistics Society* 32/1, URL: https://doi.org/10.3765/bls.v32i1.3435 [01-10-2022].

Decurtins, Alexi (2012), *Lexicon romontsch cumparativ / Vergleichendes Lexikon des Rätoromanischen. Sursilvan – tudestg*, Cuera, Societad Retorumantscha.

Decurtins, Alexi (2001), *Niev vocabulari romontsch sursilvan-tudestg. Neues rätoromanisches Wörterbuch Surselvisch-Deutsch*, Chur, Legat Anton Cadonau / Societad Retorumantscha / Verein für Bündner Kulturforschung.

Grave, Eduard / Bojanowski, Piotr / Gupta, Prakhar / Joulin, Armand / Mikolov, Tomas (2018), *Learning Word Vectors for 157 Languages*, URL: https://fasttext.cc/docs/en/crawl-vectors.html [01-10-2022].

Gusfield, Dan (1997), *Algorithms on Strings, Trees, and Sequences. Computer Science and Computational Biology*, Cambridge, Cambridge University Press.

Harris, Zellig (1954), «*Distributional Structure*», in: *Word 10* (2/3), 146–62.

Jurafsky, Daniel / Martin, James H. (2022), *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, URL: https://web.stanford.edu/~jurafsky/slp3/ed3book_jan122022.pdf [01-10-2022] ([1]2000).

Marcus, Mitchell P. (1980), *A Theory of Syntactic Recognition for Natural Language*, Cambridge (Massachusetts), MIT-Press.

Mikolov, Tomas / Chen, Kai / Corrado, Greg / Dean, Jeffrey (2013), *Efficient Estimation of Word Representations in Vector Space*, URL: https://arxiv.org/abs/1301.3781 [01-10-2022].

Salton, Gerard / McGill, Michael J. (1987), *Information Retrieval*, Hamburg etc., MacGraw-Hill.

Schirotzek, Winfried / Scholz, Siegfried (2005), *Starthilfe Mathematik: Für Studienanfänger der Ingenieur-, Natur- und Wirtschaftswissenschaften*, Wiesbaden, Teubner ([1]1995).

Spescha, Arnold (1989), *Grammatica Sursilvana*, Cuera, Casa editura per mieds d'instrucziun.

Wehrli, Eric (1997), *L'analyse syntaxique des langues naturelles. Problèmes et methodes*, Paris, Masson.

Dr. Florentin Lutz, Werdtweg 4, 3007 Bern, florentin.lutz@sunrise.ch
Prof. em. Dr. Jürgen Rolshoven, Sprachliche Informationsverarbeitung, Universität zu Köln, rols@spinfo.uni-koeln.de