Hausarbeit (Vorabgabe)
Proseminar Verarbeitung Gesprochener Sprache:

Worterkennung und Subword Segmentation:

Abstract:

Im Folgenden werden verschiedene Ansätze und Modelle der Sprachverarbeitung auf Wortebene durch Subword Segmentation verglichen, um ihre Unterschiede und die daraus entstehenden Vor- und Nachteile zu beleuchten, mit dem Ziel, festzustellen, ob die neueste Morfessor-Variante (EM+Prune) tatsächlich eine Verbesserung im Vergleich mit älteren Methoden darstellt.

Inhaltsverzeichnis:

Einführung:

Um gesprochene in geschriebene Texte (und andersherum) umwandeln oder sie übersetzen zu können, muss ein Computer in der Lage sein, Worte zu erkennen.
Es gibt dazu verschiedene Ansätze, von denen die meisten - oder genauer gesagt: die, die eine relativ hohe Trefferquote, sowie annehmbar viel Rechenzeit und Speicherverbrauch in sich vereinen - sich der Subword Segmentation bedienen, was bedeutet, dass Wörter in Untereinheiten unterteilt werden [2, S. 1-2].
Offensichtlich ist der simple Vergleich eines eingelesenen Wortes mit einem dem Computer bereits bekannten die unkomplizierteste Art, ein Wort zu erkennen; allerdings benötigt dieser Vergleich, beziehungsweise die Suche nach zum eingelesenen Wort passenden, bekannten Wörtern, Rechenzeit und das Lexikon mit den bekannten Wörtern Speicherplatz [1, S. 1].
Eine weitere Möglichkeit, die den Vorteil hat, dass weniger Speicherplatz für das Lexikon gebraucht wird, ist der Morphologische Parser [3, S. 87]: dabei werden in einem Lexikon, statt ganzen Wörtern, Wortstämme gespeichert; diese Wortstämme können nach bestimmten grammatikalischen (morphotaktischen und orthografischen) Regeln mit Affixen verbunden werden, wobei Wörter mit einer ähnlichen Bedeutung, z. B. Verben in verschiedenen Formen entstehen [3, S. 88ff.].
Dieser Ansatz ist der linguistische (nämlich der morphologische) [3, S. 79], da die Wörter als Reihung von Morphemen [3, S. 80], also den kleinstmöglichen Einheiten einer Sprache, die eine grammatikalische bzw. eine pragmatische Bedeutung haben [3, S. 81], gesehen werden.
Dieses Vorgehen ist aber ebenfalls aufwendig, da alle Wortstämme mit ihrer jeweiligen Regelzugehörigkeit gespeichert [3, S. 87], und außerdem alle Regeln softwaretechnisch umgesetzt werden müssen [3, S. 91-101], was, jeweils für eine Sprache, einen relativ hohen Aufwand für die Entwickler zur Folge hat.
Unter anderem im Licht der Kosten entwickelte sich der Ansatz der datengetriebenen "unsupervised" Segmentation; und daraus - mit dem Ziel, bessere Ergebnisse zu erzielen - die "semi-supervised" Segmentation [2, S. 2].

Zu den Vorteilen der unsupervised und der semi-supervised Segmentation gehört, dass diese Methoden für verschiedene Sprachen einsetzbar sind [2, S. 2], genauer: dass das Modell nicht für jede Sprache neu implementiert werden, sondern lediglich trainiert werden muss.
"Unsupervised" Segmentation bedeutet, dass die Daten, mit denen ein Algorithmus "trainiert" wird, zuvor nicht bearbeitet wurden [2, S. 2], während bei der "semi-supervised" Segmentation auch transkribierte Daten genutzt werden [2, S. 2].
Bei diesen Formen der Subword Segmentation werden also große (zum Teil bereits vorbearbeitete) Datenmengen benötigt, um den Algorithmus zu trainieren.

Pattern Matching und BPE:

Die einfachste Methode, um in der digitalen Sprachverarbeitung Wörter zu erkennen, ist, sie zu finden, zum Beispiel in einem Lexikon [4, S. 3]; dieses Vorgehen wird Pattern Matching genannt [1, S. 1].
Nur für eine einzige Sprache bedürfte es aber eines sehr ausführlichen Lexikons, das verhältnismäßig große Mengen an Speicherplatz verbrauchen und den Rechenaufwand steigern würde [5, S. 1], außerdem verbleibt das Problem von Wörtern, die nicht im Lexikon enthalten sind: viele Lexika enthalten 50.000 Wörter oder weniger [4, S. 1].
Eine Gruppe japanischer Wissenschaftler hat die Auswirkung von BPE (Byte Pair Encoding, ein Kompressionsalgorithmus, vorgestellt von Gage, 1994) auf die Rechenzeit beim pattern matching untersucht, mit dem Ergebnis, dass das sogenannte compressed pattern matching (die Suche in komprimierten Daten) mit einer neuen Variante von BPE weniger Rechenzeit benötigt, als die Suche in nicht komprimierten Daten [1, S. 1, 9, 11].
Zudem können Dateien mithilfe von BPE auf unter 60% ihrer ursprünglichen Größe reduziert werden [1, S. 7], wobei zusätzlich zum komprimierten Text die sogenannte Substitutionstabelle mit den Schlüsseln der einzelnen, darin vorkommenden Zeichen gespeichert wird [1, S. 4].
Das Grundprinzip, nach dem der BPE-Algorithmus funktioniert, ist relativ einfach: die zu komprimierenden Daten werden durchsucht und die zwei Zeichen (je ein Byte

groß) ermittelt, die darin am häufigsten aufeinander folgen, diese werden jeweils durch ein neues Zeichen (Byte) ersetzt und der Vorgang wird wiederholt bis entweder keine häufig vorkommenden Zeichenpaare oder keine Zeichen mehr übrig sind [1, S. 4, 5].
Die neue Variante verbessert auch die Rechenzeit die zum Komprimieren gebraucht wird, indem die Schlüssel für einen Teil des Textes ermittelt und auf den gesamten Text angewendet werden [1, S. 5].
Tatsächlich wurde BPE 2015 als Lösung für das oben erwähnte Problem der unbekannten Wörter verwendet, allerdings im Zusammenhang mit Übersetzung [4, S. 1, 2]. Dabei gibt es zwei wichtige Änderungen: erstens werden auch mehrere Zeichen durch ein neues ersetzt und zweitens werden Wörter getrennt bzw. einzeln betrachtet [4, S. 3]. Die Idee dahinter ist, unbekannte Wörter in kleinere Einheiten (z. B. Morpheme, Phoneme) zu unterteilen, deren Übersetzung bekannt ist, um sie übersetzen zu können, ohne sie zu kennen [4, S. 2].
Im Prinzip ist das der Ansatz, der auch für die morphologische Segmentation formuliert wird [2, S. 2].

Morphologische Segmentation:

Das Ziel der morphologischen Segmentation ist es, Wörter in Morpheme bzw. Morphe (Morpheme wie sie in Wörtern auftreten) zu unterteilen [2, S. 1, 2, 5]; dafür gibt es zwei Ansätze: der erste bedient sich grammatikalischer Regeln [3, Kapitel 3] und ist demnach vergleichsweise erfolgreich bzw. treffsicher, hat aber den Nachteil, dass ein System für jede Sprache aufwändig erarbeitet werden muss [5, S. 1]; der zweite basiert auf dem Training eines Algorithmus mit Daten [2, S. 5].
Die sogenannte datengetriebene morphologische Segmentation kann wiederum grob in drei verschiedene Richtungen unterteilt werden, die unsupervised, semi-supervised und supervised morphological segmentation genannt werden [5, S. 1 und 2, S. 2, 3].
Die Bezeichnungen beziehen sich auf die Daten, mit denen ein Algorithmus trainiert wird [2, S. 5].
Es gibt Listen mit Wörtern, deren Segmentation bereits vorgegeben ist, und unbearbeitete Sammlungen [2, S. 4].

Wird ein System nur mit unbearbeiteten Daten trainiert, ist das unsupervised [2, S. 2], wird es nur mit vorbearbeiteten Daten trainiert, ist es supervised [2, S. 3], und wenn beide Varianten verwendet werden, nennt man es semi-supervised [2, S. 2].
Die Forschung hat sich lange Zeit hauptsächlich mit dem Training durch unbearbeitete Daten beschäftigt, vor allem, weil diese Methoden relativ kostengünstig sind [2, S. 2].
Um die Performanz der Modelle zu verbessern, wurden in den letzten Jahren zunehmend zusätzlich auch kleine Mengen (die Anzahl der Wörter ist im drei- bis vierstelligen Bereich) vorbearbeitete Daten verwendet [2, S. 2, 5].
Es ist zu beachten, dass die morphologische Segmentation keine vollständige morphologische Analyse ist, d. h. das Ziel ist tatsächlich das Unterteilen von Wörtern in Morphe und nicht, diese grammatikalisch zu analysieren; weshalb sich das Training mit unbearbeiteten Daten überhaupt anbietet; Systeme, die eine vollständige Analyse leisten können sollen, werden üblicherweise ausschließlich mit vorbearbeiteten Daten trainiert und sind folglich mit deutlich höherem Aufwand und höheren Kosten verbunden [2, S. 4].
Außerdem ist es wichtig, zwischen Sprachen, deren Wörter häufig Morpheme bilden, und Sprachen, in denen mehrere Morpheme aneinandergereiht werden, um ein Wort zu bilden, zu unterscheiden [2, S. 4]; letztere nennt man agglutinativ [3, S. 82] und offensichtlich eignen sie sich besser für die morphologische Segmentation.
Wörter dieser Sprachen können mehrere Bedeutungen haben und demnach aus mehreren passenden Morphem-Kombinationen bestehen; eines der Systeme, die dem gerecht werden nennt sich Morfessor [2, S. 5].

Morfessor Baseline:

Der Morfessor-Algorithmus war ursprünglich unsupervised, und wurde mehrfach erweitert, unter anderem auch zu semi-supervised [2, S. 6].
Grundlage des Systems ist ein Lexikon, in dem Morphe gespeichert werden; jedes dieser Morphe hat nach bestimmten Parametern eine Auftrittswahrscheinlichkeit, von der abhängt, ob es im Lexikon bleibt [2, S. 6, 7].

Während einerseits dieses Lexikon möglichst klein sein soll, sollen andererseits alle Wörter möglichst präzise segmentiert werden [2, S. 6, 7].
Das für die semi-supervised Segmentation erweiterte (und damit jetzt) generative [2, S. 6] Modell Morfessor Baseline kennt nur eine Kategorie von Morphen (nicht mehrere, wie andere Modelle, die Wortstamm und Affixe unterscheiden) und die Morphe können nur aus Buchstaben und nicht aus kleineren Morphen bestehen; unter anderem um zu verhindern, dass zu lange Morphe im Lexikon gespeichert werden, wird jedes Morph mit Kosten belastet, die sich hauptsächlich nach der Größe des Morphs richten [2, S. 7].
Die Bewertung der Parameter läuft folgendermaßen ab: für jedes Wort (der unbearbeiteten Trainingsdatenmenge) wird iterativ nach der Segmentation mit der bestmöglichen Kostenbelastung gesucht (die vorbearbeiteten Wörter haben bereits festgelegte Werte), bis die Belastung ausbalanciert ist; wenn ein Wort segmentiert werden soll, wird die Folge von Morphen die die höchste Wahrscheinlichkeit hat mit einer Variante des Viterbi-Algorithmus berechnet [2, S. 7].
Für die semi-supervised Variante können, abgesehen von denen im Lexikon, weitere Parameter genutzt werden, um den Einfluss von vorbearbeiteten und unbearbeiteten Daten, mit dem Ziel, die Segmentation noch präziser zu machen, anzupassen [2, S. 7, 8, 13].
Morfessor gehört also zu den Lexikon-basierten Ansätzen (Trainingsziel ist das Lernen von Morphen als Lexikon) und den generativen Methoden (es geht um die Generierung von Wörtern und der passenden Segmentation) [2, S. 12].

Morfessor EM+Prune:

Morfessor EM+Prune ist die neueste Morfessor-Variante (März 2020): EM steht für Expectation Maximization und Prune für Pruning [5, S. 1, 2].
Der EM-Algorithmus wird als Alternative zum Viterbi-Algorithmus genutzt, um die bestmöglichen Werte für Parameter zu finden; dabei gibt es allerdings ein Problem: es ist nicht ratsam einerseits das Lexikon mit Morphemen zu bearbeiten, und gleichzeitig die Parameter zu ändern; das liegt daran, dass die Morphe mit Kosten belastet sind und deshalb die Belegung des Lexikons nicht geändert werden

kann, ohne dass sich die gesamte Kostenbelastung ändert, was wiederum den EM-Algorithmus stört [5, S. 2].
Dieses Problem wird mithilfe von Pruning gelöst: ein sogenanntes seed lexicon mit - anders als bei Morfessor Baseline - Subeinheiten von Wörtern, wird dazu wiederholt gekürzt, die Belegung ist danach unveränderlich [5, S. 2].
Eine weitere Änderung betrifft die Vorverteilung der Wahrscheinlichkeiten, dabei gibt es mehrere Varianten, die sich mehr oder weniger von Morfessor Baseline unterscheiden [5, S. 3, 4].

Vergleich:

Mit der Erweiterung von Morfessor Baseline von unsupervised auf semi-supervised, verbessert sich die Präzision der Segmentation [2, S. 20], sowohl die Fehlerrate bei den falschen Segmentationen, als auch bei den nicht erkannten Morph-Grenzen verringert sich [2, S. 22].
Eine Variante von Morfessor EM-Prune segmentiert wiederum um einiges präziser als Morfessor Baseline, wohingegen Morfessor Baseline, in Bezug auf die Parameter, effektiver angepasst werden kann [5, S. 1, 7].
Größere Lexika gehören ebenfalls zu den Vorteilen von Morfessor Baseline [5, S. 6, 7].
Auch innerhalb der Gruppe Morfessor EM+Prune gibt es Unterschiede: je nachdem, ob vor dem eigentlichen Pruning redundante Strings entfernt werden (pre-pruning) können im geprunten seed lexicon Strings fehlen [5, S. 7].
EM+Prune bietet außerdem eine bessere Lösung für das Kosten-Gleichgewicht des Lexikons, durch weniger Fehler in der Trainingsphase, was für präzisere Segmentation sorgt [5, S. 1, 7].

Fazit:

Der neue Morfessor-Trainings-Algorithmus stellt definitiv eine Verbesserung dar.
Das feste Lexikon ist zwar ein Nachteil, fällt aber nicht so sehr ins Gewicht, wie die verbesserte Segmentation und das wirkungsvollere Training bzw. die sinnvollere Berechnung.

Literaturverzeichnis:

[1] Byte pair encoding: a text compression scheme that accelerates pattern matching. M. Takeda, A. & T. Shinohara, Y. Shibata, T. Kida, S. Fukamachi, S. Arikawa. ()
[2] A Comparative Study of Minimally Supervised Morphological Segmentation. T. Ruokolainen, O. Kohonen, K. Sirts, S.-A. Grönroos, M. Kurimo und S. Virpioja. (2015)
[3] Speech and Language Processing. D. Jurafsky, J. H. Martin. (2nd Edition, 2009)
[4] Neural Machine Translation of Rare Words with Subword Units. R. Sennrich, B. Haddow, A. Birch. (2016)
[5] Morfessor EM+Prune: Improved Subword Segmentation with Expectation Maximization and Pruning. S.-A. Grönroos, S. Virpioja, M. Kurimo. (2020)