Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Kontextgebundene Optimierung von Edit-Distanzmaßen zwecks Entity Linking

Exposé zur Bachelorarbeit

-

eingereicht von: Fabian Hohenstein
geboren am: 02.07.1999
geboren in: Berlin

Gutachter/innen: Prof. Dr. Ulf Leser

eingereicht am: .....

1 Kontext und Problemstellung

1.1 String Similarity im allgemeinen

Die Bestimmung von Ähnlichkeiten zwischen verschiedenen Zeichenketten ist ein zentraler Bestandteil unserer multimedialen Welt. Viele Datenbankanwendungen verwenden Ähnlichkeitsmaße, um gespeicherten Text oder multimediale Objekte abzurufen. Von Data Mining über Information Retrieval bis hin zu Knowledge Discovery gewinnt die Ähnlichkeit von Zeichenketten zunehmend an Bedeutung [13].

Domänenübergreifende Problemstellungen wie die Überprüfung der Qualität gesammelter Daten sowie deren Integration greifen in vielen Anwendungsbereichen auf Ähnlichkeitssuche zurück, ebenso wie spezialisiertere Bereiche der Computerlinguistik, Bioinformatik [20] und IT-Sicherheit [8].

1.2 Präzisionsonkologie

Die Präzisionsonkologie ist ein Teilgebiet der Krebsforschung, das sich mit der individuellen molekularen Beschaffenheit betroffener Patienten befasst. Ziel ist es, individuell besser auf den Patienten und seine Bedürfnisse zugeschnittene Entscheidungen zu treffen und mit auf vorliegende Krebszellen zugeschnittenen Krebstherapien besser helfen zu können als es mit allgemeineren Ansätzen möglich wäre. [19] Das notwendige Wissen ist jedoch typischerweise dezentral in einer Vielzahl spezialisierter Datenbanken verteilt, deren Integration durch heterogene Benennungen von Krebsarten und Medikamentennamen erschwert wird. [18]

Bevor die verteilten Daten zusammengeführt werden können, ist daher ein Normalisierungsschritt erforderlich, um synonym verwendete Begriffe zu vereinheitlichen. Diese können sich typischerweise nicht auf Kontextinformationen aus vollständigen Sätzen natürlicher Sprachen stützen, da die Daten zu den referenzierten Entitäten in Form von Datenbankspalten vorliegen [10].

2 Aktueller Forschungsstand und verwandte Arbeiten

2.1 String Normalization Library PREON

Das Problem der Stringnormalisierung wurde bereits mehrfach in der Literatur behandelt und mögliche lösungsansätze vorgestellt

Der Lehrstuhl für Bioinformatik an der HU-Berlin arbeitete im Rahmen des PREDICT Projektes mit dem Institut für Pathologie der Charité an einer Softwarelösung, die es Ärzten ermöglichen soll, während der Behandlung von Krebserkrankungen schnell und intuitiv auf die große Menge an bereits vorhandenem Wissen zuzugreifen [1]. Zur Normalisierung und Integration medizinischer Daten wie Tumortypen und Medikamentennamen im PREDICT Projekt dient preon [9]. Bei PREON handelt es sich um eine Python-Bibliothek, die zur Normalisierung von Bezeichnungen für Krebsarten und

Krebsmedikamente dienen soll um automatisiert durch Stringmatching ein Mapping von Entitäten (Krebtypen und Medikamentennamen) auf Synonyme zu bilden [10].

PREON nutzt verschiedene Matchingverfahren in einer Reihenfolge aufsteigenden Rechenaufwands. Vor diesen Matching-Schritten erfolgt eine Vereinheitlichung der eingelesenen Daten um Effizienz zu gewinnen. Findet ein Schritt keine passenden Kandidaten wird der nächst komplexere Schritt ausgeführt.

0. Vereinheitlichung der Eingabe-Strings
1. exaktes Matching
2. Token-basiertes Matching
3. Edit-Distanz-basiertes Matching mit Einheitskosten

[10]

2.2 Contextuelle Edit Distanzen

Moreau et al. untersuchten die Anpassung der Gewichtung von Edit-Operationen basierend auf benachbart auszuführenden Operationen [6]. Die Arbeit ist weniger relevant, da sie Kosten der Operationen dynamisch anpasst und somit keine festen Gewichtungen im Vorfeld berechnet werden können.

2.3 Gewichtungen anhand von Tastenabständen

Gueddah et al. untersuchten die Verwendung von Zeichenabständen auf der Tastatur zur Gewichtung von Edit-Operationen, basierend auf der Annahme, dass die häufigsten Fehler Tippfehler sind, die in der Nähe der beabsichtigten Taste auftreten [11]. Diese Arbeit ist für unseren Kontext weniger relevant, da wir davon ausgehen, dass Abweichungen in den Schreibweisen von Medikamenten- und Krebsnamen weiteren Faktoren unterligen welche eine solche Gewichtung nicht abbilden würde.

2.4 Lautbasierte Kostenbewertung

Karhila et al. untersuchten 2019 die Gewichtung von Edit-Operationen anhand artikulatorischer Merkmale [14]. Diese Arbeit ist für unser Projekt weniger relevant, da wir, wie bereits bei den Tippfehlern erwähnt, davon ausgehen, dass Abweichungen in der Schreibweise von Medikamenten- und Krebsnamen nicht notwendigerweise auf Transkriptionsfehler zurückzuführen sind.

2.5 Neurale Netze anstelle diskreter Gewichte

Libovický et al. untersuchten 2021 den Ersatz diskreter Gewichtungsfunktionen von Edit-Distanzen durch Werte aus einem vortrainierten neuronalen Netz, um eine differenzierbare Verlustfunktion zu erhalten [15]. Dies ist für unser Projekt weniger relevant, da

2.6 Linear or SVM Regression Framework

Wang et al. untersuchten die Anwendung von Regressionsmodellen (Lineare und SVM-Regression) zur Optimierung von Gewichten in endlichen Automaten [21]. Dies ist für den Kontext dieser Arbeit jedoch nicht relevant, da der Nutzen endlicher Automaten durch ihre Funktion einnehmbarer Zustände nicht direkt auf eine statische Kostenfunktion mit (2d) Matrix-Lookup übertragbar ist.

2.7 Bounded Weighted Edit Distance

Kociumaka et al. untersuchten 2023 Möglichkeiten zur Beschleunigung der Berechnung gewichteter Edit-Distanzen [3]. Da hierbei jedoch nur der Algorithmus selbst im Fokus steht und nicht die erforderlichen Gewichtsfunktionen oder deren Generierung, trägt diese Arbeit nicht zur Lösung des hier behandelten Problems bei.

2.8 Expectation Maximization

Ristad und Yianilos untersuchten die Verwendung stochastischer Edit-Distanzen in Verbindung mit Methoden der Expectation Maximization. Da bei Medikamenten- und Krebsnamen formale Strukturvariationen wahrscheinlicher erscheinen als das gehäufte Auftreten von zufälligen Tippfehlern, ist der durch probabilistische Modelle verursachte Overhead für unsere Zwecke nicht gerechtfertigt. Darüber hinaus bieten EM-Algorithmen durch ihre iterative Optimierung ein direkteres Konvergenzverhalten als genetische Algorithmen, sind aber anfälliger für das Steckenbleiben in lokalen Optima. [17]

2.9 SMILES-based compound similarity

Choi, Oh untersuchten 2023 die Möglichkeit der Optimierung von Editierdistanzgewichten, um Vergleiche zwischen Zeichenketten zu ermöglichen, die im SMILES-System (Simplified Molecular Input Line Entry System) kodiert wurden. Dabei verwendeten sie sowohl das Paradigma der erschöpfenden Suche als auch die (letztlich als überlegen angesehene) genetische Programmierung [5]. Obwohl diese Arbeit nur auf die Anpassung der drei für Levensthein üblichen Gewichte abzielt, ohne die individuellen Abhängigkeiten der Buchstaben untereinander zu berücksichtigen, ist ihr Ansatz vermutlich auch für die Optimierung feinerer Gewichtsfunktionen relevant.

2.10 Musikerkennung mit GAs und Probabilistischen Algorithmen

Habrard et al. untersuchten den Einsatz stochastischer und genetischer Methoden zur Optimierung von Edit-Distanzen. für die Musikerkennung mit dem expliziten Ziel, interpretierbare Gewichte zu erzeugen. Obwohl musikalische Klangfolgen wahrscheinlich auf anderen Wahrscheinlichkeitsverteilungen basieren als medizinische Begriffe, zeigt dieser Artikel die Verwendung von genetischen Algorithmen zur Generierung von Gewichten für eine zeichensensitive Editierdistanz. [12]

3 Hintergrund und Zielsetzung

3.1 Designentscheidungen PREONs

Die von PREON verwendeten String-Vergleichsalgorithmen verwenden Ähnlichkeitsmaße (exaktes Matching, edit distance & token-based distance). Diese wurden zwar an biomedizinischen Entitätsnamen evaluiert, die zugrundeliegenden Algorithmen haben jedoch keinen inhärenten Kontextbezug zu der zu testenden Domäne. Potentiell nutzbare fachspezifische strukturelle Besonderheiten von Arzneimitteln oder Krebsbezeichnungen werden nicht berücksichtigt oder gar im Normalisierungsschritt entfernt.

3.2 Ziel der Neuentwicklung

Ziel dieser Bachelorarbeit ist die Untersuchung des Optimierungspotenzials **gewichteter** Edit-Distanzen für ein spezifisches Begriffsfeld, besonders Medikament- und Krebsnamen. In wie weit existieren kontextspezifische Worteigenschaften, welche gesichtete Edit-Distanzen erkennen und ausnutzen können, um eine genauere Suche zu ermöglichen? In diesem Sinne sollen zeichenabhängige Gewichte für Editoperationen generiert werden, die die semantische Ähnlichkeit der einzelnen Begriffe besser abbilden.

4 Grundlagen und Begrifflichkeiten

Eine Zeichenkette (engl. String) ist eine endliche Folge von einzelnen Zeichen eines zugrunde liegenden Alphabets Σ. Die Länge einer Zeichenkette s wird mit $|s|$ angegeben. Ein Substring, der mit dem i-ten Zeichen beginnt und n lang ist, kann durch die Notation $s(i, n)$ angegeben werden. $s(i)$ bezeichnet das i-te Zeichen [20].

4.1 String Matching

Sollen Existenz oder Position eines bestimmten Strings (sog. Query String) aus einer Menge von Strings bestimmt werden, so gibt es bereits für die Art der Suche verschiedene Möglichkeiten. Die naheliegendste Variante dieses Suchproblems ist der Exact String Match, bei welchem nur nach exakt gleichen Zeichenketten gesucht wird.
Effiziente Möglichkeiten zur Lösung dieses Problems (Dictionary Problem) sind u.a. Hashing oder Suchbäume. Dieses String-Matching-Problem löst jedoch nicht das Problem des Auffindens ähnlicher Zeichenfolgen, wie es bei Suchen unter Berücksichtigung von Schreib- oder Einlesefehlern oder allgemein ähnlichen Wörtern auftritt. Diese Art von Problem wird als Approximatives Dictionary Problem bezeichnet und durch approximatives String Matching gelöst.

Zur Bewertung der Ähnlichkeit zweier verglichener Zeichenketten ist eine geeignete Funktion erforderlich, von denen es verschiedene Arten gibt [13].

4.1.1 Tokenisierung

Unter Tokenisierung versteht man die Zerlegung von Eingabestrings in kleinere Einheiten, die so genannten Tokens. Diese Tokens können beliebige Teilstrings wie Wörter eines Textstrings oder q-Gramme einzelner Wortstrings sein. Jaccard-Ähnlichkeitskoeffizient, Kosinus-Ähnlichkeit und Dice-Koeffizient sind drei Beispiele für Maße die die Ähnlichkeit von Token-mengen quantifizieren und so auch für Stringähnlichkeit verwendet werden können. [22]

- **Jaccard-Ähnlichkeitskoeffizient:** Jaccard-Ähnlichkeitskoeffizient $JAC(r, s) = \frac{|r \cap s|}{|r \cup s|} = \frac{|r \cap s|}{|r|+s|-|r \cap s|}$ (wobei die Mengen r, s den tokenisierten Mengen zugrundeliegender der Strings R, S entsprechen). Tokens können im Falle von Sätzen einzelne Wörter sein, oder auch q-Gramme. [22]

4.1.2 Character-based similarity

Zeichenbasierte Ähnlichkeitsmaße messen den Abstand von Zeichenketten durch Zählung abweichender Zeichen [22].

- **Hamming Distance:** Eine mögliche Metrik zur Messung der Ähnlichkeit zweier Strings ist die Hamming-Distanz. Sie misst nicht übereinstimmende Zeichen an allen Positionen der Zeichenketten. Dieses Distanzmaß wird vor allem in der Datenübertragung zur Erkennung von Übertragungsfehlern verwendet. [22]

- **Edit Distanz** Wie die Hamming-Distanz ist die Edit-Distanz ein zeichenbasiertes Ähnlichkeitsmaß. Edit Distanzen nutzen die minimale Anzahl von String-Verändernden-Operationen, die einen Quellstring in einen Zielstring transformieren als Maß für die Ähnlichkeit dieser. Je weniger Operationen benötigt werden, desto ähnlicher sind die Zeichenketten. Die Menge erlaubter Operationen sowie ihre "Kosten"variieren von Variante zu Variante. Die **Levenshtein Distanz** verwendet beispielsweise Einfügen, Löschen bzw. Substitution eines einzelnen Zeichens. Hierbei gelten für jede Operation einheitliche Kosten [13].

- **Sonderfall: Gewichtete Edit Distanz:** Während in einigen Fällen einheitliche Kosten ausreichen, erfordern viele praktische Anwendungsfälle eine Gewichtung, bei der jeder Editieroperation ein Gewicht zugewiesen wird, das von den beteiligten Zeichen des Alphabets abhängt [7] [4]. Das Finden der optimalen Gewichte für eine Aufgabe kann die Performance des verwendeten Abstandsmaßes erheblich verbessern [5].

- **Stochastische Edit-Distanz** Unterliegen die Editoperationen einem zufälligen Prozess mit einer zugrundeliegenden Wahrscheinlichkeitsfunktion, können sie als Zufallsvariablen behandelt werden und wir nennen das Maß Stochastischen Edit-Abstand. Ist jene verteilung unbekannt kann sie durch ein geeignetes Model gelernt werden. [16]

4.2 Sequence Alignment

Für den Vergleich biomolekularer Sequenzen wie DNA, RNA oder Proteine sind mehrere Algorithmen bekannt. Diese Vergleichsalgorithmen beruhen auf gewichteten Edit-Distanzen, die je nach Anwendungsgebiet unterschiedlichen Ursprungs sind. Ohne solche angepassten Gewichte, d.h. mit einheitlichen Gewichten, wäre es schwierig, den erzeugten Ähnlichkeiten eine biologische Bedeutung zu entnehmen.

Beispielsweise werden PAM-Matrizen entwickelt, indem mehrere eng verwandte Proteine beobachtet und die Anzahl aller vorkommenden Vertauschungen von Aminosäuren für alle möglichen Aminosäurepaare gesammelt werden. Für jede Tauschmöglichkeit wird die bedingte Wahrscheinlichkeit eines symmetrischen Tausches zwischen diesen Aminosäuren berechnet und in einer symmetrischen Matrix strukturiert [2].

4.3 Genetische Algorithmen

Ein genetischer Algorithmus ist die algorithmische Nachahmung natürlicher Mechanismen der Vererbung von Merkmalen an nachfolgende Generationen gepaart mit Selektionsmechanismen. "Günstige"Gene überleben, während ungünstige Gene aussterben. Der Algorithmus hält die zu optimierenden Parameter in einem Satz von "Chromosomen"(ein Chromosom besteht aus mehreren Genen). In jeder Generation werden Chromosomen zur Fortpflanzung ausgewählt (je nach Selektionsdruck kann die Präferenz für "fittere"Chromosomen variieren). Aus je zwei Elternchromosomen werden dann Kinder gebildet (arithmetisches Mittel der Elterngene). Die neuen Kinder verdrängen nun die schwächeren Chromosomen aus der Population [5].

4.4 Expectation Maximisation Algorithmen

5 Methodik

5.1 Gewichtsoptimierung für Gewichtete Edit-Distanzen

Im Rahmen dieser Arbeit soll eine Python-Bibliothek geschrieben werden, die es ermöglicht, die Gewichte einer gewichteten Edit-Distanz auf Basis gegebener Datensätze zu optimieren.

5.1.1 Datenbeschaffenheit und deren Auswirkungen

Die Beschaffenheit der zu untersuchenden Zeichenketten hat einen erheblichen Einfluss auf die Auswahl und Effektivität der Ähnlichkeitsfunktionen. Relevant sind die Länge der Zeichenketten, die Größe des verwendeten Alphabets sowie die Struktur der Zeichenketten in sich. Im Rahmen dieser Arbeit sollen die bereits in Preon verwendeten Datensätze genutzt werden.

- Die **Stringlänge** in den vorliegenden Datensätzen ist mit durchschnittlich 9 bis 30 Zeichen relativ kurz (im Vergleich zu Aminosäure- oder Proteinsequenzen).

  - Hohe Komplexität von Edit-Distanzen fällt kaum ins Gewicht (skaliert mit Stringlänge)

- Das **Alphabet** der verwendeten Datensätze aus dem Preon Projekt umfasst 145 Zeichen
  - Es werden einzelne Zeichen in einem Wort und nicht Wörter in einem Text untersucht. Token-basierte Ähnlichkeitsmaße verlieren durch die zeichenweise Verarbeitung bei einer Token-länge von 1 einen Großteil der vorhandenen Information über die Struktur der zu vergleichenden Zeichenketten.

- Die **Struktur** der vorkommenden Zeichenketten ergibt sich häufig aus spezifischen medizinischen und chemischen Nomenklaturen und nur teilweise aus den Besonderheiten natürlicher Sprachen wie Englisch und Latein.
  - Substitutionsmatrizen wie PAM und BLOSUM sind auf ein festes Alphabet und biologische Strukturen ausgerichtet, nicht auf die Besonderheiten medizinisch/biologischer Begriffe.

5.1.2 Parameter

Die zu optimierenden **Parameter** umfassen zeichenabhängige Substitutionskosten sowie Kosten für Einfüge- und Löschoperationen (ebenfalls Zeichenabhängig). Gespeichert werden diese in drei Matritzen (eine $n * n$ sowie zwei $n * 1$ Matrizen mit $n = |\Sigma|$)

Hyperparameter sind unter anderem das Einbezogene Alphabet sowie von den Trainingsmethoden abhängende Eigenheiten wie z.B. die Anzahl der Generationen im Falle genetischer Optimierung. Interessant zu Betrachten wäre darüber hinaus wie sich der Umfang des Alphabets auf die Effektivität der Gewichte auswirkt. Sind neben Groß- und Kleinbuchstaben (distinct), arabischen Ziffern und ASCII-Basiszeichen auch Griechische Buchstaben und ungewöhnlichere Sonderzeichen wie ™ relevant, und hat im Falle des Verzichts auf diese Zeichen die Art der Normalisierung starke Auswirkungen? Hat die ersatzlose Streichung Vorteile gegenüber einer Ersetzung (z.B. $\Delta \rightarrow D$)?

5.1.3 Methoden zur Generierung & Optimierung der Gewichte für die Weighted Edit-Distance operation

Ziel: Methoden zur Bestimmung der Gewichte entwickeln und evaluieren.

- **Exhaustive Search:** Bezeichnet die vollständige (erschöpfende) Durchsuchung des Parameterraums zur Optimierung der Gewichte. Da der Parameterraum aus mindestens $128x130$ unabhängigen Variablen besteht, ist selbst bei Beschränkung des Zahlenraums der Variablen auf natürliche Zahlen innerhalb eines hinreichend kleinen Intervalls, eine vollständige Suche im Parameterraum extrem zeitaufwendig.

- **Genetische Programmierung** Die Verwendung eines genetischen Algorithmus, der durch Versuch-und-Irrtum so lange Parameter (Gewichte) mutiert, bis eine gegebene Fitnessfunktion eine hinreichende "Fitness"(in diesem Fall einen hinreichend großen F1-Wert) liefert.

5.1.4 Validierung & Evaluation

Evaluationskriterien: Analog zu PREONs Evaluation wird in dieser Arbeit die Messung von Precision, Recall, und F1 Score verwendet um den Erfolg der Gewichtung zu bestimmen und mit Preons in Relation zu setzen.

Literatur

[1] Predict - umfassende datenintegration zur verbesserung onkologischer therapien, 2021.

[2] S. Aluru, editor. *Handbook of computational molecular biology*, volume 9 of *Chapman & Hall/CRC computer and information science series*. Chapman & Hall/CRC, Boca Raton, 2006.

[3] A. Cassis, T. Kociumaka, and P. Wellnitz. Optimal algorithms for bounded weighted edit distance.

[4] A. Cassis, T. Kociumaka, and P. Wellnitz. Optimal algorithms for bounded weighted edit distance.

[5] I.-H. Choi and I.-S. Oh. Weighted edit distance optimized using genetic algorithm for smiles-based compound similarity. *Pattern Analysis and Applications*, 26(3):1161–1170, 2023.

[6] Clément Moreau, Veronika Peralta, Patrick Marcel, Alexandre Chanson, and Thomas Devogele. Learning analysis patterns using a contextual edit distance.

[7] D. Das, J. Gilbert, M. Hajiaghayi, T. Kociumaka, and B. Saha. Weighted edit distance computation: Strings, trees and dyck.

[8] S. Dolev, M. Ghanayim, A. Binun, S. Frenkel, and Y. S. Sun. Relationship of jaccard and edit distance in malware clustering and online identification (extended abstract). *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pages 1–5, 2017.

[9] A. Ermshaus. preon (precision oncology normalization). https://github.com/ermshaua/preon/, 2024. 06.05.2024.

[10] A. Ermshaus, M. Piechotta, G. Rüter, U. Keilholz, U. Leser, and M. Benary. preon: Fast and accurate entity normalization for drug names and cancer types in precision oncology. *Bioinformatics*, 40(3):btae085, 02 2024.

[11] H. Gueddah, A. Yousfi, and M. Belkasmi. The filtered combination of the weighted edit distance and the jaro-winkler distance to improve spellchecking arabic texts. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–6, 11 2015.

[12] A. Habrard, J. Iñesta, D. Rizo, and M. Sebban. Melody recognition with learned edit distances. volume 5342, 12 2008.

[13] S. S. Hasan, F. Ahmed, and R. S. Khan. Approximate string matching algorithms: A brief survey and comparison. *International Journal of Computer Applications*, 120:26–31, 2015.

[14] R. Karhila, A.-R. Smolander, S. Ylinen, and M. Kurimo. Transparent pronunciation scoring using articulatorily weighted phoneme edit distance, 2019.

[15] J. Libovický and A. Fraser. Neural string edit distance.

[16] J. Oncina and M. Sebban. Learning stochastic edit distance: Application in handwritten character recognition. *Pattern Recognition*, 39(9):1575–1587, 2006.

[17] E. S. Ristad and P. N. Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[18] M. E. Sharp. Toward a comprehensive drug ontology: extraction of drug-indication relations from diverse information sources. *Journal of biomedical semantics*, 8(1):2, 2017.

[19] S. L. Shuel. Targeted cancer therapies: Clinical pearls for primary care. *Canadian family physician Medecin de famille canadien*, 68(7):515–518, 2022.

[20] S. Wandelt, D. Deng, S. Gerdjikov, S. Mishra, P. Mitankin, M. Patil, E. Siragusa, A. Tiskin, W. Wang, J. Wang, and U. Leser. State-of-the-art in string similarity search and join. *SIGMOD Rec.*, 43:64–76, 2014.

[21] M. Wang and C. D. Manning. Spede: probabilistic edit distance metrics for mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, page 76–83, USA, 2012. Association for Computational Linguistics.

[22] M. Yu, G. Li, D. Deng, and J. Feng. String similarity search and join: a survey. *Frontiers of Computer Science*, 10(3):399–417, Jun 2016.

Appendix