Implementierung und Evaluierung des Random Indexing Wortvektormodells

Modulprojektbericht Computerlinguistische Techniken

Humanwissenschaftliche Fakultät
Department Linguistik

vorgelegt von: Simon Bross
Matrikelnummer: 809648

Berlin, den 13.03.2023

1. Einleitung

Wortvektormodelle dienen der semantischen Repräsentation von Wörtern mithilfe von Kontextvektoren in einem (definierten) Vektorraum und finden Anwendung in der Information Retrieval, Wortbedeutungsdisambiguierung sowie in der Textklassifikation. Das ihr zugrundeliegende Verfahren stützt sich auf die von Zelig Harris 1954 aufgestellte distributionelle Hypothese, die besagt, dass Wörter mit ähnlichen Bedeutungen dazu tendieren, in semantisch ähnlichen Kontexten aufzutreten. Ausgehend von diesem Prinzip entstand das prominente Zitat „*You shall know the meaning of a word by the company it keeps*“ [2] des Linguisten John Firth.

Trotz ihrer Anwendungserfolge leiden traditionelle Wortvektormodelle an Effizienz- und Skalierbarkeitsproblemen, die auf die hohe Dimensionalität und Datenspärlichkeit zurückzuführen sind. In Kookurrenzmatrizen, die mithilfe von sich über immense Korpora bewegenden Kontextfenstern erstellt werden und dabei das gemeinsame kontextuelle Auftreten von Wörtern erfassen, gleicht die Dimension eines Wortvektors beispielsweise der Anzahl der Typen im Korpus. In der Praxis sind diese Dimensionen aufgrund fehlendem gemeinsamen Kontext von Wörtern jedoch sehr spärlich besetzt und bestehen zu 99% aus Nullen, die dadurch nicht nur keinen Informationsgehalt bereitstellen, sondern auch kostbaren Speicherplatz belegen.

Das Verfahren des Random Indexings, welches Magnus Sahlgren 2005 in seinem Papier „An Introduction to Random Indexing“ [4] vorstellte, präsentiert eine Alternativemethode, die versucht, die genannten Nachteile von Wortvektormodellen zu beseitigen. Die Kernidee beruht auf der Akkumulation von Kontextvektoren, die durch Aufsummieren von Indexvektoren innerhalb von Kontextfenstern festgelegter Größe im Korpus errechnet werden. Jedem Typen im Korpus wird dabei zur Initialisierungszeit ein einzigartiger und randomisierter Indexvektor zugewiesen, dessen Dimension im Vergleich zur Kookurrenz-Methode deutlich reduziert ist und mit einer vorgegebenen Anzahl von Dimensionen besetzt ist, die zufällige Werte aus {-1, 1} annehmen.

Das vorliegende Projekt implementiert dieses Verfahren und evaluiert die generierten Modelle auf intrinsische und extrinische Weise. Bei der intrinsischen Evaluierung werden menschliche Bewertungen zur Wortähnlichkeit und Verwandschaft zwischen Wortpaaren zu Rate gezogen, die mit Kosisnusähnlichkeitswerten von Wortvektorpaaren statistisch verglichen werden. Die extrinsische Evaluierung wird anhand von Textklassifikation von Texten aus dem Brown-Korpus exemplifiziert - sprich die Modellperformanz bei der Lösung eines Anwendungsproblemes wird bewertet.

2. Korpus und Vorverarbeitung

Als Datengrundlage für die Modellerstellungen und Evaluierung wurde das vom nltk-Package bereitgestellte Brown-Korpus verwendet. Die im Projekt implementierte Klasse 'CorpusPreprocessor' kann jedoch mit jedweder Art von Korpus umgehen, sofern die vorgegebene Struktur gegeben ist. Die Methoden des Brown-Korpus liefern bereits eine tokenisierte (Satzzeichen von Tokens separiert) und mit POS-Tags versehene Version des Korpus. Die folgenden Techniken wurden für die Vorverarbeitung angewandt:

1. *Tokenisierung*: Beim Brown-Korpus bereits gegeben. Wird eine Textdatei als Korpus übergeben, wird diese in eine Liste von Listen mit Strings verarbeitet, ergo Satz- und Worttokenisierung wird durchgeführt. Dieser Schritt ist notwendig, um beispielsweise Satzzeichen von Wörtern zu trennen und allgemein um Lexeme richtig zu identifizieren ('bald!' wäre kein gültiges Token). Hierbei werden jedoch auch Mehrwortausdrücke wie 'post office' in ihre Bestandteile separiert, womit sie nicht mehr als Einheit fungieren und das Modell unter Umständen ungenau machen.

2. *POS-Tagging (Universal Tagset)*: Beim Brown-Korpus bereits gegeben. Bei Übergabe eines Korpus ohne Wortartinformationen der Tokens (.txt oder Liste von Listen ohne POS-Tags), wird POS-Tagging durchgeführt, die für die Lemmatisierung benötigt wird. Außerdem werden alle Tokens im Korpus durch ihr POS-Tag erweitert, bspw. 'can_NOUN' und 'can_VERB', um Homographe auseinanderzuhalten - als Versuch, damit die Modellgenauigkeit zu verbessern. Polyseme wie das deutsche Wort 'Bank' mit den Bedeutungen Geldinstitut, Bankgebäude und Sitzbank können damit jedoch nicht aufgelöst werden. Dies könnte jedoch mit Wortbedeutungsdisambiguierung angegangen werden.

3. *Normalisierung von Groß- und Kleinschreibung*: Alle Tokens werden in Kleinschrift transferiert, um Lexemdopplungen (wie 'Although' und 'although'), die nur auf Groß- und Kleinschreibung basieren, zu vermeiden.

4. *Interpunktionsentfernung*: Interpunktionszeichen werden entfernt, da sie nichts zur Wortsemantik beitragen, sondern eher zur Satzsemantik.

5. *Stemming*: Nicht angewandt. Durch das radikale Reduzieren auf den Stamm ist die originale Wortart und wichtige semantische Informationen nicht mehr rekonstruierbar. Verschiedene Flexionsformen/Abwandlungen eines Lexems werden dadurch auf ein einziges Token reduziert und machen das Modell unterspezifiziert.

6. *Lemmatisierung*: Wurde angewandt, um redundante morphologische Informationen eines Wortes zu entfernen. Flexionsformen werden dadurch auf ihre Grundform reduziert und das Modell verkleinert, ohne dass die originale Wortart und relevante semantische Informationen verloren gehen.

7. *Stoppwortentfernung*: Entfernt, da sie kaum semantische Kontextinformationen liefern. Stoppwörter wie beispielsweise Artikel tragen zur Wortsemantik bei, dass es sich beim

darauf folgenden Wort um ein Nomen handelt. Diese Information wird aber bereits explizit durch das POS Tagging/Tokenerweiterung hinzugefügt. Als Stoppwörter werden zusätzlich Zahlen und nicht alphabetische Tokens betrachtet, sowie Tokens, die mit einem 'X'-POS-Tag versehen sind (falsch geschriebene Wörter, Neologismen, etc.).

Bei der Vorverarbeitung ist zu beachten, dass durch ihre Strenge Kontextverfälschungen stattfinden und den Daten dadurch Rauschen hinzugefügt wird, da Wörter plötzlich mit anderen Wörtern in einem konstruierten Fenster zusammen auftreten, welches in einem unverarbeiteten Korpus nicht der Fall sein würde. Die Kontextverfälschung kann in Teilen womöglich durch die Anpassung der linken und rechten Kontextgröße neben einem Wort ausgeglichen werden. Weitere Kontextverfälschungen entstehen dadurch, dass der Algorithmus so implementiert ist, dass Satzgrenzen durch die Fenster nicht berücksichtigt werden. Für weiterführende Vergleiche wäre es noch von Interesse, Kontextfenster nur innerhalb der Satzgrenzen zu erlauben.

3. Modelle

Die Evaluierungen wurden auf 9 verschiedenen Modellen mit den folgenden Hyperparameterumfang ausgeführt:

1. Linker Fensterkontext: Diskrete Werte zwischen 5 und 10
2. Rechter Fensterkontext: Diskrete Werte zwischen 5 und 10
3. Dimensionsanzahl: Diskrete Werte zwischen 500 und 10.000
4. Anzahl der Dimensionen, die nicht null sind: Diskrete Werte zwischen 50 und 500

4. Intrinsische Evaluierung

Für die intrinsische Evaluierung der 9 Modelle wurden Datensätze von SimLex [3] und WordSim [1] herangetragen. SimLex stellt einen Datensatz aus 999 Wortpaaren (Nomen-Nomen, Verb-Verb, Adjektiv-Adjektiv) bereit, deren *similarity* von Menschen bewertet wurde (Skala: 0-10, wobei 10 am semantisch ähnlichsten beziehungsweise semantisch komplett übereinstimmend wäre). Die zwei Datensätze von WordSim bestehen ebenfalls aus Wortpaaren (Nomen-Nomen), die jeweils von Menschen anhand von *similarity* oder *relatedness* bewertet wurden. Die WordSim Skala erstreckt sich von 0-10, wobei 10 der höchste Wert wäre und bei der *similarity* nur auftritt , sofern die Wörter des Paares identisch sind. Alle Wörter der Wortpaare wurden wie in der Korpusvorverarbeitung um ihr POS-Tag erweitert, um sie mit den Modellen kompatibel zu machen.

Mithilfe der Datensätze wird gemessen, wie gut die menschlichen Bewertungen mit den Kosinusähnlichkeitswerten (Werte von [-1,1] der Wortvektorpaaren; Paare entsprechen den Wortpaaren der Datensätzen, sofern beide Wörter im Modell vorhanden sind) aus den Modellen

korrelieren. Hierzu wird die Spearman-Korrelation verwendet: Ihr p-Wert (Korrelationskoeffizient) repräsentiert die Wahrscheinlichkeit, eine Korrelation zwischen zwei Variablen (Kosinusähnlichkeitswert und Wert der menschlichen Beurteilung) zu beobachten, die genauso stark oder stärker ist als die, die in den Testdaten beobachtet wurde. Die Null-Hypothese besagt in diesem Fall, dass keine Korrelation zwischen den beiden Variablen existiert. Liegt der p-Wert unter dem Signifikanzniveau (oft 0.05), wird die Null-Hypothese verworfen und Evidenz für die Alternativhypothese - es existiert eine Korrelation - wurde gefunden.

4.1 Ergebnisse

Modell	**SimLex (Similarity)**	**WordSim (Relatedness)**	**WordSim (Similarity)**
5_5_500_50	coeff.=-0.071, p = 0.025	coeff. =0.065, p=0.323	**coeff.=0.112, p=0.134**
6_6_1000_150	coeff.=-0.068, p=0.032	coeff.=0.057, p=0.391	coeff.=0.103, p= 0.17
7_5_1500_150	coeff.=-0.071, p=0.024	coeff.=0.07, p=0.289	coeff.=0.096, p=0.199
8_6_2500_50	coeff.=-0.07, p=0.027	**coeff.=0.071, p=0.286**	coeff.=0.084, p=0.262
10_10_3000_150	coeff.=-0.065, p=0.04	coeff.=0.049, p=0.46	coeff.=0.061, p=0.409
9_6_5000_200	coeff.=-0.076, p=0.017	coeff.=0.067, p=0.313	coeff.=0.084, p=0.264
7_7_7000_200	**coeff.=-0.078, p=0.014**	coeff.=0.056, p=0.402	coeff.=0.078, p=0.297
5_9_9000_500	coeff.=-0.067, p=0.033	coeff.=0.042, p=0.525	coeff.=0.07, p=0.353
8_8_10000_300	coeff.=-0.073 p=0.02	coeff.=0.053, p=0.423	coeff.=0.073, p=0.332

Tabelle 1: Ergebnisse der intrinischen Evaluierung von 9 Random Indexing Modellen (Namensgebung in Modell-Spalte: linkerKontext_ rechterKontext_ Dimensionen_ NichtNullDimensionen), die aus dem gesamten Brown-Korpus generiert wurden. Zur Evaluierung wurde die Spearman Korrelation (Korrelationskoeffizient; = coeff.) zwischen Vektorverhältnis (Kosinusähnlichkeit von Wortvektorpaaren) und menschlichen Ratings von Wortpaaren (mittels Datensätzen von SimLex und WordSim) berechnet.

4.2 Diskussion

Die Modelle zeigen durchgehend sehr niedrige positive (bei SimLex negative) Korrelationen mit den menschlichen Bewertungen. Die p-Werte bei SimLex liegen alle unter dem Signifikanzniveau von 0.05 und sind somit statistisch signifikant. Dies mag der Wortpaaranzahl im Datensatz geschuldet sein, die deutlich höher ist im Vergleich zu WordSim, dort deuten die sehr hohen p-Werte auf eine statistische Insignifikanz. Im Vergleich zwischen *similarity* und *relatedness* schneidet jedoch die Messung von *similarity* am besten ab. Grund dafür könnte das verwendete Verfahren sein, welches keine Kookurrenzmatrix verwendet, die die Kontextfrequenz aufsummiert und somit eher mit der Messung von *relatedness* korrelieren würde, sondern stattdessen abstrakte Repräsentationen durch das Random Indexing verwendet. Es würde sich hier anbieten, die intrinische Evaluierung auch mit den Kookurrenz-Methode durchzuführen und die Ergebnisse zu vergleichen.

Im Allgemeinen könnten das abstrakte Random Indexing Verfahren, die gewählten Hyperparamter für die Modelle sowie die strikte Vorverarbeitung, die Kontexte verfälscht und

gegebenenfalls Rauschen in die Daten einfügt, für die geringen Korrelationen verantwortlich sein. Zur Verbesserung könnten die Vorverarbeitungsschritte so implementiert werden, dass sie auch als an- und ausschaltbare Hyperparameter fungieren. Damit könnten diversifiziertere Modellvarianten getestet und eine *ablation study* durchgeführt werden, um die verschiedenen Effekte der Parameter zu ermitteln. Außerdem wird das Random Indexing zur Dimensionsreduzierung verwendet, wodurch wichtige Informationen verloren gehen können beziehungsweise nur sehr abstrakt und schlecht extrahierbar dargestellt werden. Es ist hier auch zu erwähnen, dass die Neuberechnung der Modelle mit denselben Hyperparametern zu divergierenden Ergebnissen führt, sprich die Modellperformanz schwankt trotz gleicher Hyperparameter aufgrund der randomisierten Erstellung.

5. Extrinsische Evaluierung

Zur extrinsischen Evaluierung wurden die Wortvektoren der Modelle als Features zur Darstellung von Texten verwendet, um mit ihnen einen Klassifikator zu trainieren und anzuwenden. Die 500 Texte im Brown-Korpus sind in folgende 15 Kategorien eingeteilt, die der Klassifikator vorhersagen kann: 'hobbies', 'lore', 'humor', 'romance', 'fiction', 'reviews', 'religion', 'science_fiction', 'belles_lettres', 'editorial', 'news', 'adventure', 'mystery', 'government', 'learned'. Die Training-, Validierungs- und Testingsplits sind aufgrund der Klassenaufteilung sehr unbalanciert. Ausführliche Ergebnisse pro Klasse und pro Modell liegen in der PDF im 'outputs' ZIP-Ordner vor.

5.1 Ergebnisse aller Modelle über alle Klassen

Model	Precision	Recall	F1
model_5_5_500_50	0.19	0.21	0.14
model_6_6_1000_150	0.21	0.21	0.14
model_7_5_1500_150	0.21	0.21	0.14
model_8_6_2500_50	0.21	0.21	0.14
model_10_10_3000_150	0.21	0.21	0.14
model_9_6_5000_200	0.21	0.21	0.14
model_7_7_7000_200	0.21	0.21	0.14
model_5_9_9000_500	**0.19**	**0.21**	**0.15**
model_8_8_10000_300	0.21	0.21	0.14

Tabelle 2: Overall Precision, Recall und F1 (macro) Werte der 9 verschiedenen Modelle aus der extrinsischen Evaluierung. Die Modelle wurden auf den jeweiligen Validierungssplits getestet. Das Modell 5_9_9000_500 schnitt dabei am besten ab. Auf ihm wurde zum Schluss mit dem Testingsplit getestet (letzte Tabellenreihe)

Auch die Ergebnisse der extrinsischen Evaluierung sprechen für eine niedrige Performanz und eine niedrige semantische Repräsentationsfähigkeit der generierten Modelle. Precision, Recall und F1 erreichten keine Werte über 0.21, 0.21 und 0.15. Auffallend sind nicht nur die 7 Modelle, für die die Werte exakt gleich sind, sondern auch deren F1 Werte, die unterhalb derer der Precision und Recall liegen. Obwohl die implementierten Funktionen manuell überprüft wurden, liegt der Grund dafür im Ungewissen. Möglicherweise ist aber das Klassenungleichgewicht und die Berechnung des F1 Wertes schuld, weil diese einen macro-Durchschnitt verwendet und einige Klassen keine Vorhersagen haben - also eine Precision, Recall und F1 von 0. Während in der intrinsischen Bewertung ein Korrelationsmaß verwendet wurde, wurden die Modelle in diesem Teil dazu verwendet, ein Klassifikationsproblem zu lösen. Die Wortvektoren der Modelle scheinen jedoch keine guten Features abzugeben, sprich die Texte semantisch nicht gut genug zu repräsentieren, um damit einen brauchbaren Klassifikator zu trainieren. Dies kann jedoch auch an den geringen Mengen von Text pro Klasse im Brown-Korpus liegen.

5.2 Ergebnis des besten Modells auf dem Validierunsteil

Das Modell mit den Hyperparametern 5, 9, 9000, 500 (linker Kontext, rechter Kontext, Dimensionen, Nicht-Null-Dimensionen) erzielte auf dem Testingsplit folgende Ergebnisse über alle Klassen: 0.27 (Precision), 0.25 (Recall), 0.19 (F1). Auch hier liegt der F1 Wert unüblicherweise unter der Precision und Recall, alle drei Werte stellen jedoch Maxima dar im Vergleich zu Tabelle 2. Die folgende Tabelle zeigt die drei Metriken pro Klasse:

Tabelle 3: Precision (P), Recall (R), und macro F1 score (F) für jede Klasse auf model_5_9_9000_500

Kategorie	**P**	**R**	**F**
hobbies	0.50	0.14	0.22
lore	0.27	0.30	0.29
humor	0.00	0.00	0.00
romance	0.36	0.67	0.47
fiction	0.25	0.33	0.29
reviews	0.13	0.67	0.22
religion	0.14	0.33	0.20
science_fiction	0.00	0.00	0.00
belles_lettres	0.17	0.13	0.15
editorial	0.00	0.00	0.00
news	1.00	0.33	0.50
adventure	0.00	0.00	0.00
mystery	0.00	0.00	0.00
government	0.17	0.67	0.28
learned	1.00	0.12	0.22

Die Kategorien 'science_fiction' (insg. 6 Texte im Korpus), 'editorial' (insg. 27 Texte im Korpus), 'mystery' (insg. 24 Texte im Korpus), 'adventure' (insg. 29 Texte im Korpus) und 'humor' (insg. 9 Texte im Korpus) weisen keine Vorhersagen auf. Bei 'science_fiction' und 'humor' mag dies an der sehr starken Unterrepräsentierung im Korpus liegen, bei 'editorial' an einem stark variierendem Schreibstil und semantischen Mustern in den Texten. Der Stil und die Semantik in der Kategorie 'news' (44 Texte im Korpus) ist mutmaßlich einheitlicher und einfacher zu erkennen, was sich in dem höchsten F1 score widerspiegelt (0.5). Dasselbe gilt für die Kategorie 'romance' (F1 von 0.47) in denen sich wahrscheinlich bestimmte lexikalische und semantische Muster zum Thema Liebe wiederholen. Zur Performanzverbesserung bieten sich dieselben Vorschläge an, die bereits in 4.2 ausgeführt worden sind, allerdings sollten auch ausbalanciertere Daten in größeren Mengen für diese Aufgabe verwendet werden, um aussagekräftigere Ergebnisse und unverfälschte F1 Werte zu erhalten.

Literatur

[1] Eneko Agirre u. a. "A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, Juni 2009, S. 19–27. URL: https://aclanthology.org/N09-1003.

[2] J. Firth. "A Synopsis of Linguistic Theory 1930-1955". In: *Studies in Linguistic Analysis*. reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow. Philological Society, Oxford, 1957.

[3] Felix Hill, Roi Reichart und Anna Korhonen. *SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation*. 2014. DOI: 10.48550/ARXIV.1408.3456. URL: https://arxiv.org/abs/1408.3456.

[4] Magnus Sahlgren. "An introduction to random indexing". In: *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE)*. Copenhagen, Denmark, 2005.