Verwendung von Wissensgraphen zur inhaltlichen Ergänzung kleinerer Textkorpora

Hagen, Thora
thora.hagen@uni-wuerzburg.de
Julius-Maximilians-Universität Würzburg

Problemstellung

Die Verwendung von statischen oder dynamischen Word Embeddings wie FastText (Bojanowski et al. 2016) oder BERT (Devlin et al. 2019) hat die Verarbeitung natürlicher Sprache auch im Bereich der digitalen Geisteswissenschaften wesentlich verbessert. Allerdings setzen diese Verfahren voraus, dass man zu ihrem Training über ein sehr großes Textkorpus verfügt, zum Beispiel Wikipedia oder das OSCAR Korpus (Suárez et al. 2020), mit mehreren Gigabyte Umfang. Viele DH-Projekte können aber nur auf sehr viel kleinere Textmengen zurückgreifen. Andererseits beschäftigen sich viele DH-Projekte mit kultureller Überlieferung, die schon seit längerer Zeit erforscht wird, so dass etwa Wörterbücher oder andere strukturierte Nachschlagewerke vorliegen. Dieses Paper diskutiert, wie man ein Word Embedding wie FastText auf sehr kleinen Textmengen trainieren kann und durch die Hinzufügung von Wörterbüchern als Wissensgraphen eine deutlich verbesserte abstrakte semantische Repräsentation des Korpus erreichen kann.

Wissensgraphen oder Knowledge Graphen sind eine Form der Informationsrepräsentation, bei der systematisch Aussagen in der Form Subjekt-Prädikat-Objekt (Tripel) dargestellt werden. Die Informationen im Graph können einer spezifischen Domäne angehören oder auch Allgemeinwissen insgesamt abbilden. Das automatische Umwandeln einer lexikalisch-semantischen Ressource (LSR) – das können beispielsweise Wörterbücher oder Enzyklopädien sein – in einen Wissensgraphen ist nicht zuletzt durch die eher offen gehaltene Definition eines Wissensgraphen unproblematisch, da die Einträge in LSR häufig bereits in einer Art Tripel-Struktur organisiert sind. Die Erstellung des Graphen aus einer LSR reicht über einfache regelbasierte Verfahren (Chodorow 1985) über Clustering Methoden (Oliveira und Gomes 2011), bis hin zu der Verwendung von Sprachmodellen, wobei hier hauptsächlich der Anspruch besteht, Tripel aus Fließtext zu extrahieren (siehe z.B. Yang et al. 2020).

Anreicherung von Word Embeddings durch Wissensgraphen

Die Forschung im Bereich der natürlichen Sprachverarbeitung und Knowledge Graphen hat gezeigt, dass Word Embeddings sowie auch Sprachmodelle von strukturiertem Wissen profitieren können. Der Ansatz ist getrieben von der Intuition, dass einige semantische Relationen in Form von Fließtext selten ausgedrückt werden, da sie für Menschen offensichtlich sind, z.B. "er aß die *gelbe* Banane" oder "Friedrich Schiller war eine Person" (anstatt die Berufsbezeichnungen zu nennen). Lediglich durch distributionelle Semantik würden sich solche Beziehungen nicht unbedingt in darauf basierenden Modellen niederschlagen.

Um statische Embeddings mit Informationen aus einem Wissensgraphen anzureichern, gibt es hauptsächlich zwei Ansätze:

1. nachträgliches Angleichen der Embeddings an den Wissensgraph („Retrofitting“; Faruqui et al. 2015) oder
2. Konkatenation von Knowledge Graph Embeddings und Word Embeddings sowie anschließende Dimensionsreduktion („Fusion“; Thoma et al. 2017).

Ebenfalls möglich ist das parallele Trainieren der Embeddings auf Fließtext und Wissensgraph (Xu et al. 2014), welches sich allerdings vor allem gegenüber des erstgenannten Ansatzes aufgrund der höheren benötigten Rechenleistung nicht durchgesetzt hat.

Beim Retrofitting werden bereits vortrainierte Word Embeddings im Nachhinein durch einen Wissensgraphen angepasst. Dabei werden die unmittelbaren Nachbarschaften im Graphen ausgenutzt: iterativ werden die Wortvektoren so angepasst, dass die Distanzen zu den direkten Nachbarn im Graphen und gleichzeitig die Distanz zum jeweiligen ursprünglichen Wortvektor minimiert werden.

Die Fusion-Methode verfolgt einen anderen Ansatz: zuerst werden Embeddings auf Basis des Wissensgraphen berechnet. Ähnlich wie bei Word Embeddings auch wird dabei jeder Entität im Graph ein Vektor zugeordnet, welcher durch die Nähen zu den anderen Vektoren aus dem Graph die Bedeutung der Entität abbildet. Populäre Ansätze sind zum Beispiel TransE (Bordes et al. 2013) oder RotatE (Sun et al. 2019). Bei TransE werden die Vektoren so trainiert, dass die Summe aus Subjekt- und Prädikatvektor möglichst nah an dem Vektor des Objekts liegt. Viele andere Verfahren bauen auf der Idee auf, so auch RotatE – hier wird die Beziehung zwischen Objekt und Subjekt durch eine Rotation im Vektorraum über das Prädikat (anstelle der Summe) abgebildet. Für die Fusion werden dann ebenfalls vortrainierte Word Embeddings mit den Embeddings der Entitäten aus dem Graph konkateniert. In einem zweiten Schritt werden dann die konkatenierten Embeddings auf die gewünschte Dimension reduziert, zum Beispiel durch eine Principle Component Analysis (PCA). Dabei können die verschiedenen Embeddings je nach Anwendungsfall unterschiedlich gewichtet werden. Für Knowledge Base Completion zum Beispiel, also die automatische Vorhersage neuer Relationen in einem Graph, eignen sich Fusionsembeddings, bei welchen der Wissensgraph stärker gewichtet wurde, besser (Thoma et al. 2017).

Das hier dargestellte Konzept kann ebenso für das Anreichern von Sprachmodellen wie BERT verwendet werden, denn gerade Sprachmodelle benötigen so wie das Trainieren von Word Embeddings auch viele Textdaten, um eine Sprache angemessen abbilden zu können. Auch hierbei gibt es verschiedene Möglichkeiten; darunter beispielsweise das Einhängen der Tripel-Informationen in den Fließtext einhergehend mit dem Anpassen des Attention Mechanismus für das Pre-training (Liu et al. 2019) oder das Erstellen eines gänzlich neuen Textes mittels zufälliger Pfade aus dem Graphen, welcher via eigener Adapter in das Pre-training des Sprachmodells integriert wird (Lauscher et al. 2020).

Methodik

Im Folgenden sollen exemplarisch Ergebnisse für das Anpassen von statischen Word Embeddings mithilfe eines Wissensgraphen auf Basis einer kleinen Textmenge dargestellt werden. Um eine kleine Domäne zu simulieren wurde aus dem Deutschen OSCAR Korpus eine Menge an zufälligen Sätzen so ausgewählt, dass etwa 20MB (ca. 3.6M Tokens, 47.000 Types) an Text daraus entstanden sind. Mit diesen Daten wurde dann ein 300-dimensionales Word Embedding Modell mit FastText trainiert. Für den Wissensgraph wurde GermaNet (Hamp und Feldweg 1997, Henrich und Hinrichs 2010) herangezogen. Ähnlich zu dem englischen WordNet werden in GermaNet semantische Beziehungen zwischen Wörtern verzeichnet (Synonyme, Hyponyme etc.). Es ist deshalb zu erwarten, dass die angepassten Word Embeddings vor allem Wortähnlichkeiten besser abbilden können. Obwohl sowohl statische als auch kontextualisierte Embeddings für das Experiment verwendet werden können, wurden hier die statischen Embeddings gewählt, da diese für das Abbilden von semantischen Beziehungen immer noch genauso gut geeignet sind (Ehrmanntraut et al. 2021).

Speziell für diese Evaluation wurden deshalb mehrere Datensätze ausgewählt, welche Wortähnlichkeiten und Wortverwandschaften prüfen: Schm280 (Köper et al. 2015), SimLex-999 (Leviant und Reichart 2015), ZG222 (Zesch und Gurevych 2006) und Gur65 sowie Gur350 (Gurevych 2005). Bei allen Datensätzen besteht jede Testinstanz aus einem Wortpaar und einer manuell annotierten Wertung der Wortähnlichkeit. Da nicht immer alle Wörter einer Testinstanz in den Word Embeddings gefunden werden, werden nicht alle Instanzen bei der Evaluation berücksichtigt. Die Anzahl der tatsächlich verwendeten Testinstanzen je Testset können in Tabelle 1 eingesehen werden.

Tab. 1: Performanz der angepassten Word Embeddings auf den ausgewählten Datensätzen (Spearman Korrelationen zwischen den Kosinus Ähnlichkeiten der Wortvektorpaare und der menschlichen Bewertungen). Für das FastText Modell sind zusätzlich die Standardabweichungen von jeweils 15 Durchläufen gegeben.

	SimLex-999	Schm280	ZG222	Gur65	Gur350
# Instanzen	825	242	120	49	237
FastText	0,224 (0,004)	0,495 (0,01)	0,299 (0.01)	0,320 (0,03)	0,653 (0,01)
Retro_all	0,267	0,512	0,291	0,490	0,607
Retro_syns	0,253	0,487	0,273	0,374	0,652
Fusion	0,250	0,484	0,347	0,426	0,666
Retro+Fusion	0,278	0,537	0,337	0,497	0,660

Sowohl Retrofitting als auch der Fusions-Ansatz wurden hier getestet. Faruqui et al. (2013) verwenden beim Retrofitting nur ein Subset der Relationen: nur Synonyme oder Synonyme zusammen mit Hyponymen und Hyperonymen. Für dieses Experiment wurden ebenfalls zwei verschiedene Relationssets ausgewählt: 1) alle Relationen von GermaNet (*Retro_all*) und 2) nur Synonym-Tripel (*Retro_syns*). Für beide Fälle wurden nur jene Tripel auch verwendet, bei welchen Subjekt sowie Objekt in den vortrainierten Word Embeddings enthalten waren. Somit wurden für 1) etwa 80.000 Tripel für das Retrofitting verwendet während bei 2) nur etwa 10.000 verwendet wurden. Um die Tripel zu erstellen wurden nur die Lemmas und nicht die Synset-Struktur aus GermaNet verwendet; also Wörter aus mehreren Synsets werden demselben Vektor in den Word Embeddings zugeordnet, ohne dass eine Disambiguierung stattfindet. Für die Implementierung wurde eine optimierte Version des Retrofitting Algorithmus von Lengerich et al. (2017) herangezogen.

Für die Fusions-Methode wurden RotatE Embeddings (Implementierung von Zhu et al. (2019)) mit einer Dimension von 128 trainiert. Nachteil dieser Methode ist, dass die Mehrheit der Wörter aus dem Vokabular der Word Embeddings kein Gegenstück in den Entitäten von GermaNet haben (etwa 62%). Fehlende GermaNet Vektoren wurden in diesem Experiment deshalb durch zufällig bestimmte Vektoren innerhalb der Grenzen des GermaNet RotatE Vektorraumes erstellt. Damit ist sichergestellt, dass alle Modelle auf Basis des gleichen Vokabulars beurteilt werden. Die Word und Knowledge Graph Embeddings wurden nicht weiter gewichtet; die Dimensionsreduktion wurde mit einer PCA vorgenommen.

Berechnet wurden die Spearman Korrelationen zwischen den menschlichen Bewertungen und der Kosinus-Ähnlichkeiten der Vektoren der Wortpaare (siehe Ergebnisse in Tabelle 1). Für das vortrainierte FastText sind die Mittelwerte sowie die Standardabweichungen der Spearman Korrelationen aus 15 identisch trainierten Modellen angegeben, um etwaige Schwankungen aufgrund der nicht-deterministischen Modellerstellung anzuzeigen.

Für die Auswertung wurden außerdem *Retro_all* und das Fusions-Modell miteinander kombiniert um ein Ensemble-Modell zu präsentieren und einen Konsens zwischen den Modellen zu bilden. Im Ensemble-Modell werden deshalb die Kosinus Ähnlichkeiten beider Modelle gemittelt und für die Auswertung herangezogen.

Auswertung und Diskussion

Anhand der Ergebnisse zeigt sich, dass auf allen Datensätzen das Anpassen der Word Embeddings mit GermaNet zu einer Verbesserung der Performanz führt. Vor allem das Ensemble-Modell *Retro_all+Fusion* erzielt dabei konsistent bessere Resultate. Insbesondere für die Repräsentation von Wortähnlichkeiten (bzw. Synonymie in SimLex-999) erscheint es lohnenswert, die Anpassung der Word Embeddings durch GermaNet vorzunehmen. Speziell beim Retrofitting fallen die Ergebnisse des auf dem gesamten Relationsbestand von GermaNet optimierten Modells besser aus als nur bei den Synonymen. Bemerkenswert ist trotzdem, dass *Retro_syns*, welches auf einem vergleichsweise kleinem Set aus Tripeln abgestimmt wurde, ebenfalls schon in manchen Fällen Fortschritte erzielen kann.

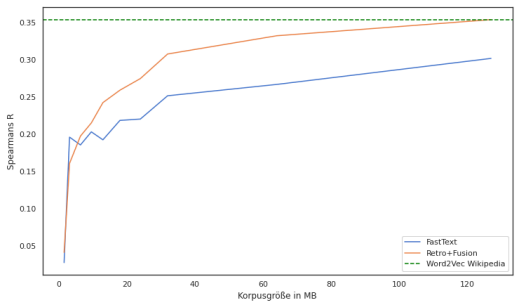

Abb. 1: Performanz des Retro_all+Fusion Modells im Vergleich zu den nicht angepassten FastText Äquivalenten auf dem SimLex-999 Datensatz mit zunehmender Korpusgröße. Als Baseline ist die Performanz des auf Wikipedia trainierten Word2Vec Modells von Leviant und Reichart (2015) gegeben.

Um deutlich zu machen, wie sich die Korpusgröße insgesamt auf die Ergebnisse auswirkt, wurde noch ein weiteres Experiment umgesetzt. Dafür wurden zunächst verschieden große Textsamples aus dem OSCAR Korpus generiert, um dann jeweils ein normales FastText sowie ein durch GermaNet angepasstes Modell

auf verschieden großen Korpora zu vergleichen. Für das Fitting wurde *Retro_all+ Fusion* gewählt; die Evaluation wurde auf SimLex-999 durchgeführt. Die Ergebnisse sind in Abbildung 1 dargestellt.

Hauptsächlich drei Beobachtungen können aus diesem Experiment abgeleitet werden. Erstens gibt es trotzdem ein unteres Limit für die Korpusgröße, ab der das Fitting der Vektoren zu keiner Verbesserung führt, vermutlich da das FastText Modell allein schon zu wenig Informationen enthält. Hier sind es etwa 10MB für das OSCAR Korpus; allerdings ist es möglich, dass für tatsächlich domänenspezifische Korpora dieses Limit weiter unten angesetzt ist, da das Vokabular von OSCAR sich über alle Domänen erstreckt. Ein kleineres Vokabular, gegeben durch eine spezifische Domäne, würde hier vielleicht auch unterhalb der 10MB ein sinnvolles FastText Modell trainieren können.

Die zweite Beobachtung ist, dass im Falle eines 20MB großen Korpus durch das Fitting eine Performanz eines etwa dreimal so großen Korpus erzielt wird: bei 64MB zeigt das normale FastText Modell eine Performanz von 0,27. Drittens lässt sich anhand der angezeigten Baseline zeigen, dass ein Korpus der Größe 120MB mithilfe des Fittings bereits genauso erfolgreich ist wie ein auf Wikipedia (aktuell etwa 13GB) trainiertes Word2Vec Modell.

Für das hier durchgeführte Experiment wurden keine Hyperparameter optimiert, sowohl für das Trainieren der FastText Embeddings als auch für das Anpassen mit beiden Ansätzen. Durch weiteres Anpassen der Lernrate beim Retrofitting oder bei der Wahl des Knowledge Graph Embedding Algorithmus und der Anzahl der Dimensionen für diese können möglicherweise noch bessere Ergebnisse erzielt werden. Auch die Auswahl der Tripel aus einem Graphen oder die Auswahl des Graphen an sich kann eine entscheidende Rolle spielen; prinzipiell kann diese je nach Anwendungsfall für die Word Embeddings unterschiedlich ausfallen. Wenn mit den Embeddings beispielsweise das Erkennen von Entitäten eher im Fokus steht, ist es denkbar, Tripel aus DBpedia (Lehmann et al. 2015) oder Wikidata zum Verbessern der Vektoren zu verwenden, da dort hauptsächlich Personen und Orte verzeichnet sind. Geht es eher um die Erkennung von Part-of-Speech, so kann die Zuhilfenahme eines Wörterbuches, welches morphologische Informationen zu den Wörtern beinhaltet, nützlicher sein.

Dieses Paper zeigt insgesamt, dass es sich lohnt, eine zur Verfügung stehende lexikalisch-semantische Ressource in den Erstellungsprozess von Word Embeddings zu integrieren; hier demonstriert anhand der Erkennung semantischer Wortähnlichkeiten in der deutschen Sprache. Vor allem dann, wenn wenig Daten vorhanden sind, um ein Forschungsvorhaben in einer speziellen Domäne durchzuführen, können diese zusätzliche Ressourcen ausgenutzt werden um ein Korpus inhaltlich anzureichern und somit das Trainieren eines Word Embeddings Modells unterstützen. Typische Domänen können zum Beispiel ein historisches Korpus, Dialekte, Pidgins und andere Arten von Sprachvariation oder auch ein ganz spezifisches Genre sein. Vor allem also für Germanisten, die auf Grundlage einer eher textarmen Domäne mit quantitativen Methoden arbeiten möchten (sei es beispielsweise für das Erkennen von Bedeutungsveränderungen von Wörtern mithilfe von Embeddings innerhalb einer solchen Domäne), kann das Anreichern von Textkorpora mit Wissensgraphen und eines der hier vorgestellten Verfahren von Interesse sein.

Bibliographie

Bordes, Antoine / Usunier, Nicolas / Garcia-Durán, Alberto (2013): „Translating Embeddings for Modeling Multi-relational Data“, in: *Advances in neural information processing systems* 2787-2795.

Bojanowski, Piotr / Grave, Edouard / Joulin, Armand / Mikolov, Tomas (2016): „Enriching Word Vectors with Subword Information“, ArXiv:1607.04606.

Chodorow, Martin S. / Byrd, Roy J. / Heidorn, George E. (1985): „Extracting Semantic Hierarchies from a Large On-Line Dictionary“, in: *Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics* 299–304.

Devlin, Jacob / Chang, Ming-Wei / Lee, Kenton / Toutanova, Kristina (2019): „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“, ArXiv:1810.04805.

Ehrmanntraut, Anton / Hagen, Thora / Jannidis, Fotis / Konle, Leonard (2021): „Type- and Token-based Word Embeddings in the Digital Humanities“, in: *CEUR Workshop Proceedings 2989*.

Faruqui, Manaal / Dodge, Jesse / Jauhar, Sujay K. / Dyer, Chris / Hovy, Eduard / Smith, Noah A. (2015): „Retrofitting word vectors to semantic lexicons“, in: *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference* 1606–1615.

Gurevych, Iryna (2005): „Using the structure of a conceptual network in computing semantic relatedness“, in: *International conference on natural language processing* 767–778.

Hamp, Birgit / Feldweg, Helmut (1997): „GermaNet - a Lexical-Semantic Net for German“, in: *Automatic information extraction and building of lexical semantic resources for NLP applications.*

Henrich, Verena / Hinrichs, Erhard W. (2010): „GernEdiT-The GermaNet Editing Tool“, in: *ACL (System Demonstrations)* 19–24.

Köper, Maximilian / Scheible, Christian / Schulte im Walde, Sabine (2015): „Multilingual Reliability and ‚Semantic‘ Structure of Continuous Word Spaces“, in: *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015) -- Short Papers.*

Lauscher, Anne / Majewska, Olga / Ribeiro, Leonardo F. R. / Gurevych, Iryna / Rozanov, Nikolai / Glavaš, Goran (2020): „Common Sense or World Knowledge? Investigating Adapter-Based Knowledge Injection into Pretrained Transformers“, ArXiv:2005.11787.

Lehmann, Jens / Isele, Robert / Jakob, Max / Jentzsch, Anja / Kontokostas, Dimitris / Mendes, Pablo N. / Hellmann, Sebastian / Morsey, Mohamed / Van Kleef, Patrick / Auer, Sören / u. a. (2015): „DBpedia--a large-scale, multilingual knowledge base extracted from wikipedia“, in: *Semantic Web* 6 167–195.

Lengerich, Benjamin J. / Maas, Andrew L. / Potts, Christopher (2017): „Retrofitting distributional embeddings to knowledge graphs with funtional relations“, ArXiv:1708.00112.

Leviant, Ira / Reichart, Roi (2015): „Separated by an Uncommon Language: Towards Judgment Language Informed Vector Space Modeling“, ArXiv:1508.00106.

Liu, Weijie / Zhou, Peng / Zhao, Zhe / Wang, Zhiruo / Ju, Qi / Deng, Haotang / Wang, Ping (2020): „K-BERT: Enabling language representation with knowledge graph“, in: *Proceedings of the AAAI Conference on Artificial Intelligence* 2901-2908.

Oliveira, Hugo G. / Gomes, Paulo (2011): „Automatic discovery of fuzzy synsets from dictionary definitions“, in: *Twenty-Second International Joint Conference on Artificial Intelligence.*

Ortiz Suárez, Pedro J. / Romary, Laurent / Sagot, Benoît (2020): „A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages“, in: *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics.

Sun, Zhiqing / Deng, Zhi-Hong / Nie, Jian-Yun / Tang, Jian (2019): „RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space“, ArXiv:1902:10197.

Thoma, Steffen / Rettinger, Achim / Both, Fabian (2017): „Towards holistic concept representations: Embedding relational knowledge, visual attributes, and distributional word semantics“, in: *Lecture* Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10587 LNCS: 694–710.

Xu, Chang / Bai, Yalong / Bian, Jiang / Gao, Bin / Wang, Gang / Liu, Xiaoguang / Liu, Tie Yan (2014): „RC-NET: A general framework for incorporating knowledge into word representations“, in: *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management* 1219–1228.

Yang, SungMin / Yoo, SoYeop / Jeong, OkRan (2020): „DeNERT-KG: Named Entity and Relation Extraction Model Using DQN, Knowledge Graph, and BERT“, in: *Applied Sciences* 10.

Zesch, Torsten / Gurevych, Iryna (2006): „Automatically creating datasets for measures of semantic relatedness“, in: *Proceedings of the workshop on linguistic distances* 16–24.

Zhu, Zhaocheng / Xu, Shizhen / Qu, Meng / Tang, Jian (2019): „GraphVite: A High-Performance CPU-GPU Hybrid System for Node Embedding“, in: *The World Wide Web Conference* 2494–2504.