Clustern von Relationen zwischen Wortvektoren[1]

Dennis Ulmer

Computerlinguistisches Abschlusskolloquium
Insitut für Computerlinguistik

15. Dezember 2015

[1] Arbeitstitel

Gliederung

Distributionelle Semantik & Knowledge Graph Completion

Distributional hypothesis

Words that occur in the same contexts tend to have similar meanings (Harris 1954)

↪ Trainieren von Wortvektoren ($\rightarrow$ `word2vec`)

- $vector(King) - vector(Man) + vector(Woman) \approx vector(Queen)$

Knowledge Graph Completion

Knowledge Graph Completion

"Knowledge graphs encode structured information of entities and their rich relations. [...] [A] typical knowledge graph [...] is usually far from complete. **Knowledge graph completion** aims at predicting relations between entities under supervision of the existing knowledge graph" (Lin et al. 2015)

↪ In semantischen Vektorraummodellen manifestieren sich manche Relationen durch einen ähnlichen Vektor zwischen Wortpaaren (→ TransE)

Idee

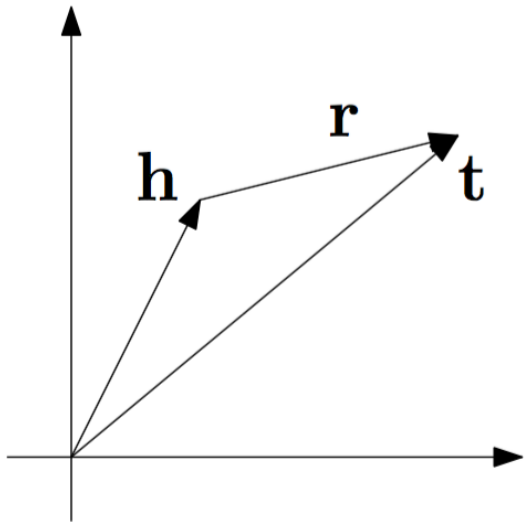
→ Trainieren von 1:1-, 1:N-, N:1- und N:N-Relationen in einem semantischen Vektorraum (Lin et al. 2015)

→ Auffinden von Relationen innerhalb des Vektorraums fürs Deutsche (diese Bachelorarbeit!)

TransE

- Wortvektoren $h, t \in \mathbb{R}^k$
- Relationstripel: (h, r, t), sodass $h + r \approx t$
- Kostenfunktion: $f_r(h, t) = \|h + r - t\|_2^2$

$\rightarrow$ Probleme: Modellieren von *1:N*-, *N:1*- und *N:N*-Relationen



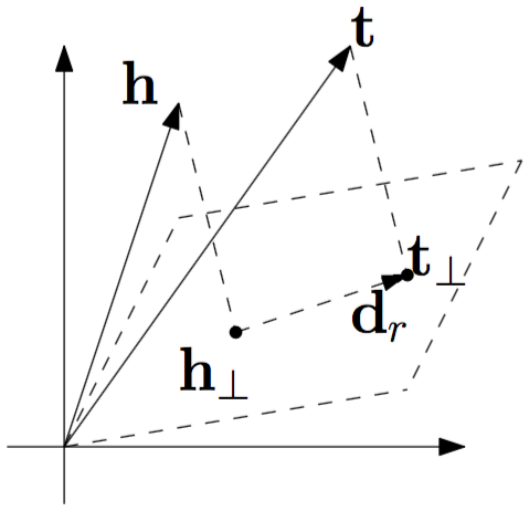
TransH (Wang et al. 2014)

- Idee: Projektion der Wortvektoren auf relationsspezifische Ebene, um alle Arten von Relationen modellieren zu können
- Projektion von h und t mit Normalvektor w_r mit $\|w_r\|_2 = 1$:

  $h_\perp = h - w_r^\top h w_r$

  $t_\perp = t - w_r^\top t w_r$
- $f_r(h,t) = \|h_\perp + r + t_\perp\|_2^2$

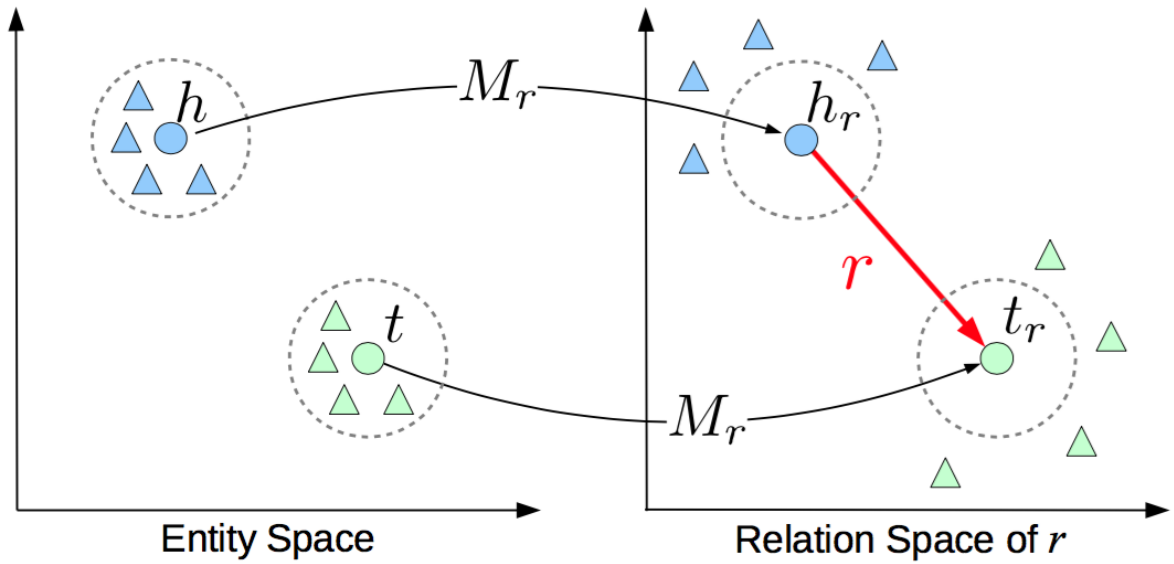
$\rightarrow$ Probleme: Verschiedene (für Relationen wichige) Aspekte von Entitäten im Vektorraum werden nicht unterschieden



TransR

- Idee: Trennung von Entitäts- und Relationsraum
- Relationstripel (h, r, t) mit $h, t \in \mathbb{R}^k$ und $r \in \mathbb{R}^d$, wobei $k \neq d$ sein kann
- Projektionsmatrix für jede Relation $M_r \in \mathbb{R}^{d \times k}$:

  $h_r = hM_r; t_r = tM_r$
- (Alte) Kostenfunktion $f_r(h, t) = \|h_r + r - t_r\|$



Cluster-based Trans-R (CTransR)

→ Problem: Eine Relation wird verschiedenen "Lesarten" nicht gerecht

- Idee: Entitäten einer Relation nach Versatz $(h - t)$ clustern
- Danach: TransR mit clusterspezifischem r_c trainieren
- $f_r(h, t) = \|h_{r,c} + r_c - t_{r,c}\|_2^2 + \alpha\|r_c - r\|_2^2$

Diese Arbeit

- Nicht alle Typen von Relationen funktionieren in einem semantischen Vektorraum

→ Menschliche Vorauswahl (mit menschlichem Bias!)

- Idee: Entitätspaare nach Versatz (und anderen Features) clustern und Cluster auf Zugehörigkeit zu Relation prüfen
  - Input: Deutsche Wortvektoren
  - Output: Cluster aus Wortpaaren, die einem Relationstypen zuzuschreiben sind

Hypothese

- Trennung von Entitäts- und Relationsraum (hier allerdings nur **ein** Relationsraum)
- Wortpaar (h, t), z.B. $(\textit{Paris}, \textit{Frankreich})$ oder $(\textit{Paris}, \textit{Blumentopf})$
- Mapping des Wortpaares in den Relationsraum:

  $\phi(h, t) \mapsto \tilde{r}_{h,t}$

  $h, t \in \mathbb{R}^k, \tilde{r}_{h,t} \in \mathbb{R}^d$, wieder $k \neq d$ möglich

Hypothese

Cluster aus nahe beieinanderliegende $\tilde{r}_{h,t}$ gehören zu einer Relation

- Anspruch an $\phi(\cdot, \cdot)$: Zu einer Relation gehörenden Wortpaare sollten im Relationsraum möglichst nahe beieinander landen
- Mögliche Features für Mappingfunktion $\phi(\cdot, \cdot)$:
  - Richtung [2]
  - Länge
  - ...

$\rightarrow$ Offene Fagen:

- Welche Features für $\phi(\cdot, \cdot)$ übernehmen?
- Welche Features funktionieren? Welche nicht?
- (Wie TransE) *1:N*-, *N:1*- und *N:N*-Relationen?

[2] $\tilde{r}_{h,t} = t - h$ (= TransE!)

Decow-Korpus

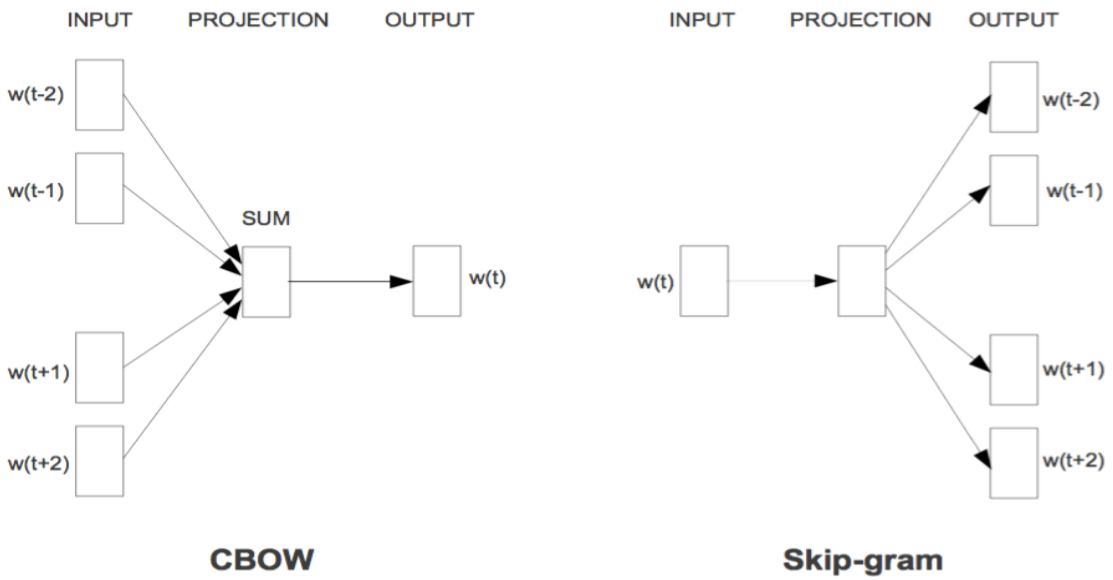
= **DE co**rpus from the **w**eb

- Texte von deutschsprachigen Internetseiten
- Annotation mit PoS-, NE-Tags sowie Morphologie und Lemmata
- Außerdem: Diverse Meta-Informationen zu einzelnen Sätzen
- 21 Teile mit insgesamt $>$ 11,5 Mrd. Tokens in $>$ 600 Mio. Sätzen

Freebase

- Von Community gepflegte Graphbasierte Datenbank → Entitäten sind Knoten, Relationen sind Kanten
- In mehreren Sprachen verfügbar, durch Google API und Abfragesprache MQL abfragbar
- Aus Freebase: FB40K-Datenset (335.350 englische Relationstripel)

word2vec



- C-Tool zum Trainieren von Wortvektoren (*Word embeddings*)
- Parameter:
  - Skip-gram- oder Bag-of-words-Ansatz
  - Lernalgorithmus
  - Dimensionalität
  - Größe des Kontextfensters
  - Subsampling häufiger Wörter
  - ...

→ Optimierung der Parameter auf Korpusausschnitt?

Vorbereitung

- Deutsche Äquivalente für Relationsenden der FB40k-Daten abfragen (erfolgreich für $\sim$ 60 %) $\rightarrow$ 204.016 deutsche Tripel
- Extrahieren aller NEs und ihrer Satz-IDs aus dem Decow-Korpus $\rightarrow >$ 31 Mio. NEs
- Vorkommen von Mehrwort-NEs in Decow mit Unterstrich verbinden
- Trainieren der Wortvektoren
- Extraktion von verwandten Entitäten zur Wortvektorevaluation

Clustering

- Clustering der $\tilde{r}_{h,t}$ im Relationsraum nach räumlicher Nähe
- Clustering-Algorithmus verwenden, der die Anzahl der Cluster nicht von Anfang an vorgibt

$\rightarrow$ Offene Fragen:

- Sinnvolle Einschränkungen für den Algorithmus?
- Welche Wortpaare machen überhaupt Sinn?
  $(\textit{Paris}, \textit{Frankreich}) \leftrightarrow (\textit{Paris}, \textit{Blumentopf})$

Evaluation

- Daten aus Decow und FB40k
  - Sind die ins deutsche übertragenen Daten aus FB40k fürs deutsche Relevant?

  → Anteil von FB40k-NEs, die auch in Decow-NEs sind

  → **88.954 %**
- Wortvektoren
  - Haben die Wortvektoren überhaupt semantische Aussagekraft?

  → Anteil von räumlich nahen Entitäten in Freebase-"Konzeptgruppen"
- Geclusterte Relationen
  - Wurden wirkliche Relationen gefunden?
  - Einschränkung fürs erste auf NEs

  → Anteil von Entitätspaaren in einem Cluster, die sich der gleichen Freebase-Relation zuordnen lassen

Fahrplan

- Vorbereitungen ⊖
  - NEs in Decow extrahieren ✓
  - Deutsche FB40k-Tripel extrahieren ✓
  - ♢ **Evaluation: Relevanz von dt. FB40k-Tripeln** ✓
  - Decow für Training vorbereiten ⊖
  - Konzeptgruppen aus Freebase extrahieren
  - Wortvektoren trainieren
  - ♢ **Evaluation: Wortvektoren**
- Mapping
  - Verschiedene Mappingfunktionen $\phi(\cdot, \cdot)$ erstellen & ausprobieren
- Clustering
  - Verschiedene Clusteralgorithmen ausprobieren
  - ♢ **Evaluation: Gefundene Relationscluster**

✓: Erledigt ⊖: In Arbeit

Literatur

- Harris, Zellig S. "Distributional structure." Word (1954).
- Levy, Omer, Yoav Goldberg, and Ido Dagan. "Improving distributional similarity with lessons learned from word embeddings." Transactions of the Association for Computational Linguistics 3 (2015): 211-225.
- Lin, Yankai, et al. "Learning entity and relation embeddings for knowledge graph completion." Proceedings of AAAI. 2015.
- Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. 2013.
- Wang, Zhen et al. "Knowledge graph embedding by translating on hyperplanes." Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014.

Sonstiges

- `word2vec`. Tool for computing continuous distributed representations of words. `https://code.google.com/p/word2vec/`
- `Freebase`. A community-curated database of well-known people, places, and things. `https://www.freebase.com`