StarSpace: Embed All The Things!

Wu et al. [2018]

Alexey Ivanov Nicolas Weber

Universität Heidelberg
Institut für Computerlinguistik
Embeddings
Katja Markert & Ines Rehbein
SoSe 19

27.06.2019

Gliederung

Einführung

- StarSpace stellt ein Modell zum Erstellen neuraler Embeddings für Entitäten dar

Einführung

- StarSpace stellt ein Modell zum Erstellen neuraler Embeddings für Entitäten dar
- Durch das Einbetten in den gleichen Vektorraum können Entitäten verschiedener Typen sinnvoll miteinander verglichen werden

Einführung

- StarSpace stellt ein Modell zum Erstellen neuraler Embeddings für Entitäten dar
- Durch das Einbetten in den gleichen Vektorraum können Entitäten verschiedener Typen sinnvoll miteinander verglichen werden
- Viele verschiedene Anwendungsmöglichkeiten mit guter Performance

Modell I

- Grundlegende Idee: Es werden Entitäten modelliert
- Entitäten lassen sich über unterscheidbare Features als bag-of-features (BOF) darstellen
- Jedem Feature wird ein d-dimensionales Embedding zugewiesen
- Erstellen eines Dictionaries F bestehend aus $\mathcal{D}$ verschiedenen Features und deren Embeddings der Länge d $\rightarrow \mathcal{D} \times d$ Matrix

$$F_{\mathcal{D},d} = \begin{pmatrix} feat_{1,1} & feat_{1,2} & \cdots & feat_{1,d} \\ feat_{2,1} & feat_{2,2} & \cdots & feat_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ feat_{\mathcal{D},1} & feat_{\mathcal{D},2} & \cdots & feat_{\mathcal{D},d} \end{pmatrix}$$

Modell II

- Zugriff auf Embedding für Feature i über F_i
- Embedding für Entität mit BOF a:

$$\sum_{i \in a} F_i$$

- Das Modell soll lernen Entitäten zu vergleichen
- Zu minimierende Loss Funktion:

$$\sum_{\substack{(a,b) \in E^+ \\ b^- \in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Modell III

- Generator für positive Paare aus E^+, abhängig von der Task
- Generator für k negative Paare aus E^-, durch negative Sampling (Mikolov et al. [2013])
- Größe von k ist ein Hyperparameter
- Ähnlichkeitsfunktion $sim(\cdot, \cdot)$, hier als Hyperparameter mit Wahl zwischen Kosinusähnlichkeit und Skalarprodukt

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Modell IV

- Loss Funktion L^{batch}:
  - Margin ranking loss:

$$L^{batch} = \sum_{b_k^- \in E^-} max(0, \mu - sim(a, b) + sim(a, b_k^-))$$

  - Negative log loss (Softmax)

$$\sum_{\substack{(a,b) \in E^+ \\ b^- \in E^-}} L^{batch}(sim(a, b), sim(a, b_1^-), \ldots, sim(a, b_k^-))$$

Modell V

- Methode zur Optimierung ist Stochastic Gradient Descent (SGD)
- Embeddings werden normalisiert
- Funktion $sim(\cdot, \cdot)$ kann zum Testen verwendet werden
  - $max_{\hat{b}} sim(a, \hat{b})$ für mögliche $\hat{b}$
  - Sortieren von Entitäten nach Ähnlichkeit
- Gelernte Embeddings können auch direkt verwendet werden

Multiclass Classification

- Bsp. Text Classification
- E^+ sind annotierte Trainingsdaten mit (a, b) als Paare von Dokumenten a und Labels b
- b^- wird aus der Menge möglicher Labels gewählt

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Collaborative Filtering-based Recommendation I

- User als Paar aus ID (Feature) und Items (BOF)
- a ist die User-ID und b ein einzelnes Item im BOF
- b^- werden aus den möglichen Items gewählt

Problem:
Da jedem User eine eigene ID zugewiesen wird, lassen sich nicht einfach neue User hinzufügen

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Collaborative Filtering-based Recommendation II

Alternative:

- b ist wieder ein einzelnes Item im BOF
- a ist keine ID, sondern die Summe der Items ohne b
- Ein neuer User kann repräsentiert werden, als Summe seiner Items

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Multi-Relational Knowledge Graphs

- Graph als Tripel (h, r, t)
- Instanzen von h,r und t sind jeweils Features in F
- Entweder: a als BOF aus h und r und $b = t$
- Oder: $a = h$ und b als BOF aus r und t
- In beiden Fällen wird b^- aus den möglichen Konzepten gewählt

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Sentence Embeddings

- Trainingsdaten sind nichtannotierte Dokumente
- a und b sind zwei Sätze aus dem gleichen Dokument
- b^- sind Sätze anderer Dokumente
- bei langen Dokumenten könnte man ein Kontextfenster einführen, um semantische Ähnlichkeit besser zu gewährleisten

$$\sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

Text Classification I

Text Classification auf drei Datensätzen:

- AG news[1]
  - 4 Klassen Text Classification
  - 120K Trainings- und 7600 Testinstanzen
  - ca. 100K Wörter und 5M Tokens
- DBpedia [Lehmann et al., 2015]
  - 14 Klassen
  - 560K Trainings- und 70K Testinstanzen
  - ca. 800K Wörter und 32M Tokens
- Yelp review Datenset aus der 2015 Yelp Dataset Challenge[2]
  - 5 Klassen
  - 1,2M Trainings- und 157K Testinstanzen
  - ca. 500K Wörter und 193M Tokens

Text Classification II

Model	AG news	DBpedia	Yelp15
BoW*	88.8	96.6	-
ngrams*	92.0	98.6	-
ngrams TFIDF*	92.4	98.7	-
char-CNN*	87.2	98.3	-
char-CRNN⋆	91.4	98.6	-
VDCNN◇	91.3	98.7	-
SVM+TF†	-	-	62.4
CNN†	-	-	61.5
Conv-GRNN†	-	-	66.0
LSTM-GRNN†	-	-	67.6
fastText (ngrams=1)‡	91.5	98.1	**62.2
StarSpace (ngrams=1)	91.6	98.3	62.4
fastText (ngrams=2)‡	92.5	98.6	-
StarSpace (ngrams=2)	92.7	98.6	-
fastText (ngrams=5)‡	-	-	66.6
StarSpace (ngrams=5)	-	-	65.3

Table 2: Text classification test accuracy. * indicates models from (Zhang and LeCun 2015); ⋆ from (Xiao and Cho 2016); ◇ from (Conneau et al. 2016); † from (Tang, Qin, and Liu 2015); ‡ from (Joulin et al. 2016); ** we ran ourselves.

Text Classification III

Training time	ag news	dbpedia	Yelp15
fastText (ngrams=2)	2s	10s	
StarSpace (ngrams=2)	4s	34s	
fastText (ngrams=5)			2m01s
StarSpace (ngrams=5)			3m38s

Table 3: Training speed on the text classification tasks.

[1] https://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
[2] https://www.yelp.com/dataset_challenge

Link Prediction I

- Freebase 15K Dataset [Bordes et al., 2013]
  - 14.951 Konzepte, davon 1.345 Relationen
  - Trainingset: 483.142 Tripel
  - Validationset: 50.000 Tripel
  - Testset: 59.071 Tripel

Link Prediction II

Metric	Hits@10 r.	Mean Rank r.	Hits@10 f.	Mean Rank f.	Train Time
SE* (Bordes et al. 2011)	28.8%	273	39.8%	162	-
SME(LINEAR)* (Bordes et al. 2014)	30.7%	274	40.8%	154	-
SME(BILINEAR)* (Bordes et al. 2014)	31.3%	284	41.3%	158	-
LFM* (Jenatton et al. 2012)	26.0%	283	33.1%	164	-
RESCAL† (Nickel, Tresp, and Kriegel 2011)	-	-	58.7%	-	
TransE (dim=50)	47.4%	212.4	71.8%	63.9	1m27m
TransE (dim=100)	51.1%	225.2	82.8%	72.2	1h44m
TransE (dim=200)	51.2%	234.3	83.2%	75.6	2h50m
StarSpace (dim=50)	45.7%	191.2	74.2%	70.0	1h21m
StarSpace (dim=100)	50.8%	209.5	83.8%	62.9	2h35m
StarSpace (dim=200)	52.1%	245.8	83.0%	62.1	2h41m

Table 4: Test metrics on Freebase 15k dataset. * indicates results cited from (Bordes et al. 2013). † indicates results cited from (Nickel et al. 2016).

Relevanz von k

K	1	5	10	25	50	100	250	500	1000
Epochs	3260	711	318	130	69	34	13	7	4
hit@10	67.05%	68.08%	68.13%	67.63%	69.05%	66.99%	63.95%	60.32%	54.14%

Table 5: Adapting the number of negative samples k for a 50-dim model for 1 hour of training on Freebase 15k.

Learning Sentence Embeddings I

- Wikipedia Dataset: [Chen et al., 2017]
  - 5.035.182 Artikel mit 9,008.962 Tokens
  - Trainingsset: 5.035.182 Artikel
  - Validationsset: 10.000 Artikel
  - Testset: 10.000 Artikel
- Trainingsablauf:
  - Wählt zufällig zwei Sätze aus einem Artikel als (a,b)
  - Wählt aus 10.000 anderen Artikeln Negativbeispiele b^-

Learning Sentence Embeddings II

- Verschiedende Modelle:
  - Trainiert auf Wortebene
  - Trainiert auf Satzebene
  - Trainiert auf Wort- und Satzebene
  - Zusammengesetzt aus 13 Modellen trainiert auf Wort- und Satzebene
- Evaluierung über SentEval[3] Tool

Learning Sentence Embeddings III

Task	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
Unigram-TFIDF*	73.7	79.2	90.3	82.4	-	85.0	73.6 / 81.7	-	-	0.58 / 0.57
ParagraphVec (DBOW)*	60.2	66.9	76.3	70.7	-	59.4	72.9 / 81.1	-	-	0.42 / 0.43
SDAE*	74.6	78.0	90.8	86.9	-	78.4	73.7 / 80.7	-	-	0.37 / 0.38
SIF(GloVe+WR)*	-	-	-	82.2	-	-	-	-	84.6	0.69 / -
word2vec*	77.7	79.8	90.9	88.3	79.7	83.6	72.5 / 81.4	0.80	78.7	0.65 / 0.64
GloVe*	78.7	78.5	91.6	87.6	79.8	83.6	72.1 / 80.9	0.80	78.6	0.54 / 0.56
fastText (public Wikipedia model)*	76.5	78.9	91.6	87.4	78.8	81.8	72.4 / 81.2	0.80	77.9	0.63 / 0.62
StarSpace [word]	73.8	77.5	91.53	86.6	77.2	82.2	73.1 / 81.8	0.79	78.8	0.65 / 0.62
StarSpace [sentence]	69.1	75.1	85.4	80.5	72.0	63.0	69.2 / 79.7	0.76	76.2	0.70 / 0.67
StarSpace [word + sentence]	72.1	77.1	89.6	84.1	77.5	79.0	70.2 80.3	0.79	77.8	0.69/0.66
StarSpace [ensemble w+s]	76.6	80.3	91.8	88.0	79.9	85.2	71.8 / 80.6	0.78	82.1	0.69 / 0.65

Table 9: Transfer test results on SentEval. * indicates model results that have been extracted from (Conneau et al. 2017). For MR, CR, SUBJ, MPQA, SST, TREC, SICK-R we report accuracies; for MRPC, we report accuracy/F1; for SICK-R we report Pearson correlation with relatedness score; for STS we report Pearson/Spearman correlations between the cosine distance of two sentences and human-labeled similarity score.

Learning Sentence Embeddings IV

Task	STS12	STS13	STS14	STS15	STS16
fastText (public Wikipedia model)	0.60 / 0.59	0.62 / 0.63	0.63 / 0.62	0.68 / 0.69	0.62 / 0.66
StarSpace [word]	0.53 / 0.54	0.60 / 0.60	0.65 / 0.62	0.68 / 0.67	0.64 / 0.65
StarSpace [sentence]	0.58 / 0.58	0.66 / 0.65	0.70 / 0.67	0.74 / 0.73	0.69 / 0.69
StarSpace [word+sentence]	0.58 / 0.59	0.63 / 0.63	0.68 / 0.65	0.72 / 0.72	0.68 / 0.68
StarSpace [ensemble w+s]	0.58 / 0.59	0.64 / 0.64	0.69 / 0.65	0.73 / 0.72	0.69 / 0.69

Table 10: Transfer test results on STS tasks using Pearson/Spearman correlations between sentence similarity and human scores.

[3] https://github.com/facebookresearch/SentEval

Fazit

- Vergleichsweise gute Ergebnisse in den durchgeführten Experimenten
- Anwendbar auf weitere Problemstellungen
- Repräsentation von Entitäten als Kombination von Features und die taskspezifische Wahl von E^+ und E^- machen das Modell flexibel

Gibt es noch Fragen?

Vielen Dank für die Aufmerksamkeit!

1. Wie kommt man von der Repräsentation einer Entität als bag-of-features zu einem Embedding der Entität?
2. Wo liegt der Unterschied in der User-Repräsentation bei Collaborative Filtering-based Recommendation (CFR) und CFR with out-of-sample user extension?
3. Was kann man bei der Link Prediction für verschiedene Werte des Hyperparameters k, bei Einschränkung der Trainingszeit auf eine Stunde, beobachten?

Literatur I

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Literatur II

Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.