sent2vec

Leon Schmidt, Lukas Reuter

Universität Heidelberg
Institut für Computerlinguistik
Embeddings
Prof. Dr. Katja Markert & Dr. Ines Rehbein
SoSe 2019

27. Juni 2019

Outline

- Motivation
- sent2vec - Algorithmus
  - Recap word2vec C-BOW
  - sent2vec
- Trainingsdaten
- Evaluation (Testdaten)
- Ausblick

Motivation

- Bisher sehr gute Resultate für Wort-Embeddings, noch bessere Resultate für Satz-Embedding
- Verschiedene Ansätze für Satz-Embeddings (Wieting et al., 2016b; Arora et al., 2017)
- Simplererer Algorithmus mit besserer Perfomance möglich?

Motivation

Heutiges Modell: sent2vec

Matteo Pagliardini, Prakhar Gupta, Martin Jaggi (2016)

> *Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features*

Recap word2vec & C-BOW

Recap word2vec & C-BOW

- word2vec berechnet Wort-Vektoren mithilfe eines neuronalen Netzwerkes
- Dabei wird über ein Korpus iteriert und Wörter und deren Kontexte werden betrachtet
- Verschieden Modelle bisher gesehen: Skip-Gram & C-BOW, GloVe, ...
- Heute wichtig: C-BOW (Continuous Bag Of Word Model)

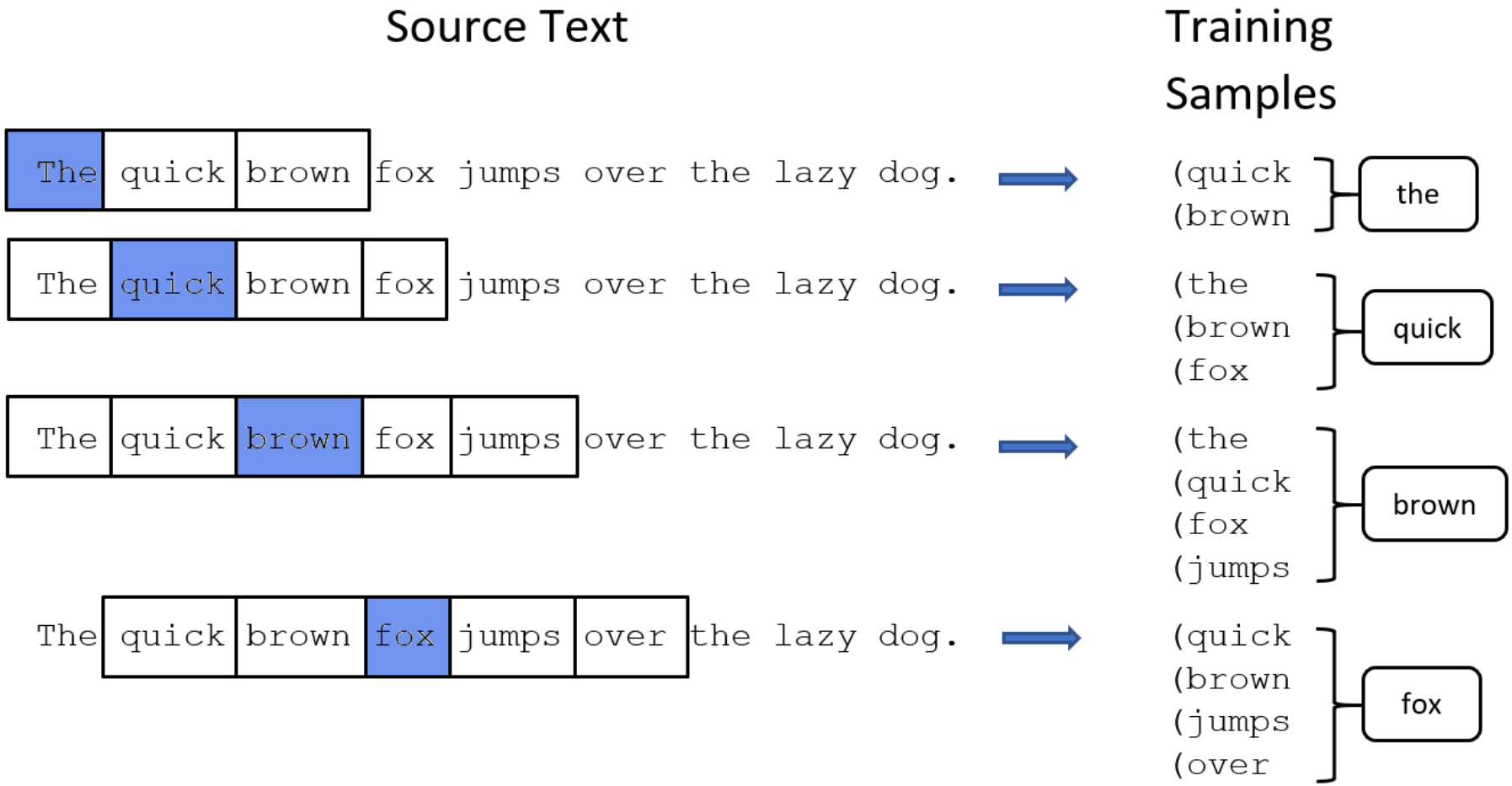
Recap word2vec & C-BOW

Skip-Gram Model:
Gegeben: center word w_c, Output: context words w_o

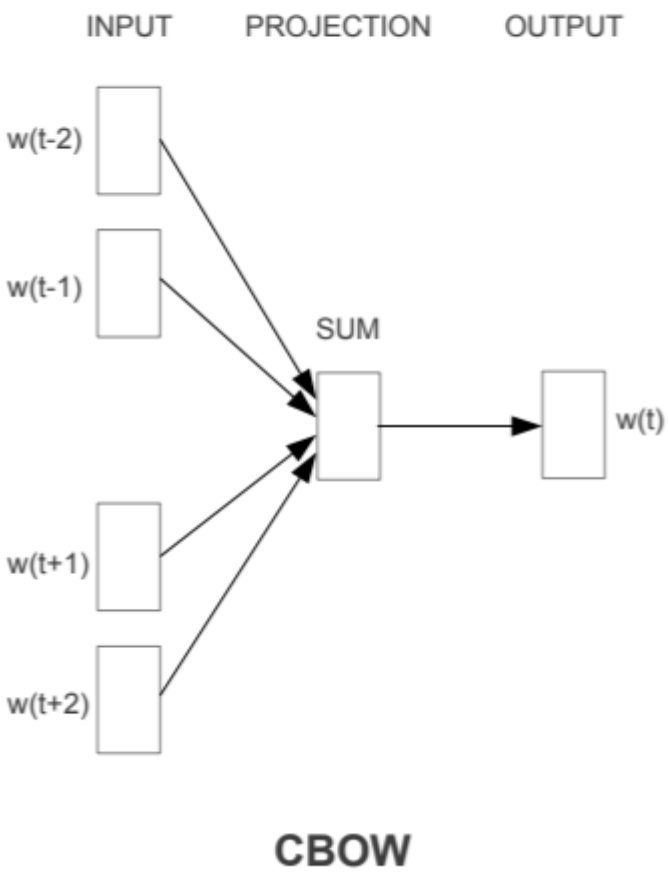
C-BOW Model:
Gegeben: context words w_o, Output: center word w_c

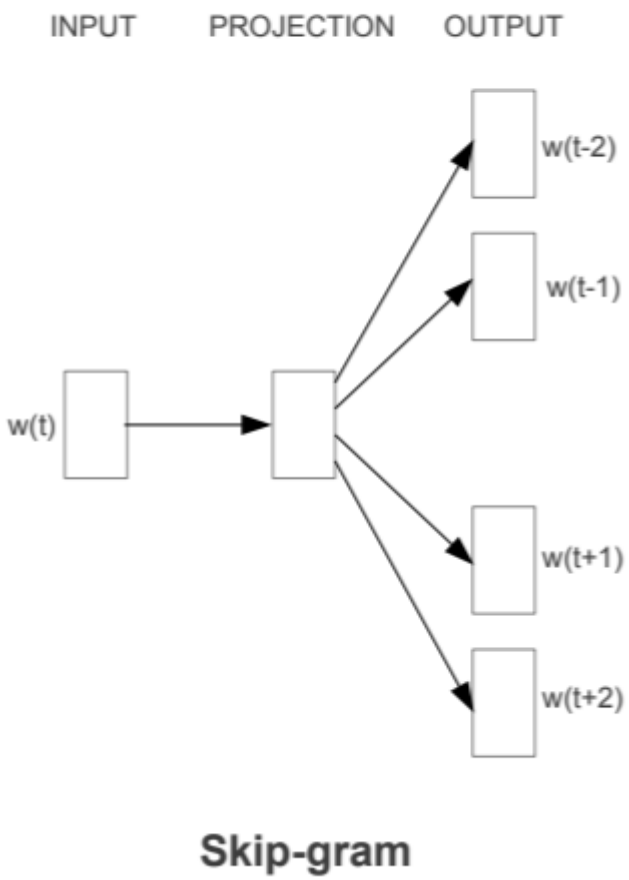
Recap C-BOW

Reminder: Äquivalent zu Beispiel von Skip-Gram



Recap word2vec & C-BOW





(Illustration von Mikolov et al., 2013)

Recap C-BOW

C-BOW Beispiel mit einem(1) Kontextwort als Input*

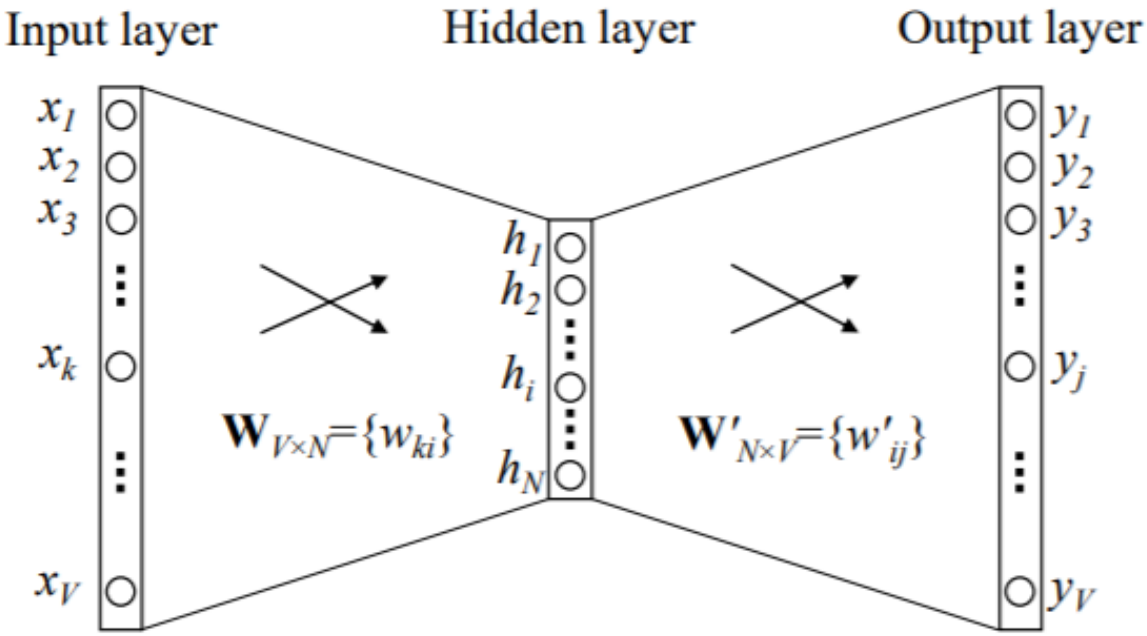


Figure 1: A simple CBOW model with only one word in the context

*Terminologie aus Papier von Xin Rong, 2016

(Illustration von Xin Rong, 2016)

Recap C-BOW

- **Input** $\{x_{k...V}\}$: One-Hot-Vektor(en) Input-Wort(e)
- **Gewichte** Input Layer –> "Hidden Layer" können als VxN Matrix W verstanden werden, wobei

  V = Vokabulargröße

  N = Anzahl Parameter *im "Hidden Layer" (frei wählbar*)*
- Jede Zeile in W kann als der jeweilige Vektor für das k-te Wort gesehen werden. Da nur ein Wert im Input Layer 1 ist und der Rest 0, wird nur eine Zeile in W ausgewählt –> **Vektor** für Kontext-Wort k
- Sei unser Vektor für das Kontextwort k: $\boldsymbol{V}_{w_k}$

*Bei C-BOW Training (Mikolov et al. 2013) N=1000

Recap C-BOW

- **Gewichte** "Hidden Layer" –> Output Layer können als NxV Matrix $W'_{NxV} = \{w'_{ij}\}$ verstanden werden.
  - In jeder Spalte j der Vektor für das Center-Wort j
- Sei unser **Vektor** für das Center-Wort j: u_{w_j}

Recap C-BOW

- **Output** $\{y_{j...V}\}$: Dimension V mit einzelnen Werten für y_j (jedes Wort j).
- Berechne Wert für jedes Output-Center-Wort:= $u_{w_j}^T v_{w_k}$, wobei
  - u_{w_j} = Vektor für Center-Wort j
  - v_{w_k} = Vektor für das Kontext-Wort k
- Softmax-Funktion gibt Wahrscheinlichkeitsverteilung für jeden Output-Wert, also für jedes Center-Wort an.

–> Output:= **softmax(**$u_{w_j}^T v_{w_k}$**)**

Recap C-BOW

C-BOW Beispiel mit mehr als einem Kontextwort als Input

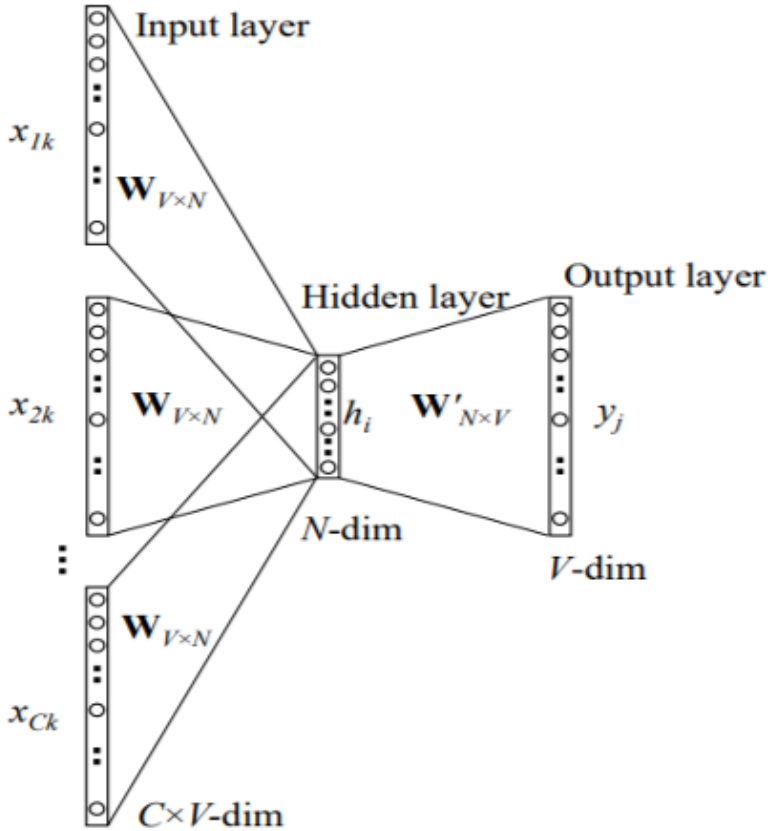


Figure 2: Continuous bag-of-word model

(Illustration von Xin Rong, 2016)

Recap C-BOW

Der Vektor h im "Hidden Layer" mit den Parametern der Anzahl N ist nun der Durchschnitt der Wortvektoren v_{w_k} aus dem Input und den jeweiligen Gewichtematrizen W_k Also:

$$\begin{aligned} h &= \frac{1}{C} W^T (x_1 + x_2 + ... + x_C) \\ &= \frac{1}{C} (v_{w_1} + v_{w_2} + ... + v_{w_C})^T, wobei \end{aligned} \tag{1}$$

C = Anzahl der Wörter im Kontext
$w_1, w_2, ..., w_C$ = Wörter im Kontext
v_w = Vektor für ein Kontext-Wort w

Recap C-BOW

Damit wäre der Output nun **softmax(**$u_{w_j}{}^T h$**)**

Recap C-BOW

- Trainieren des Modells mithilfe von Backpropagation
  - Jedes Output-Wort y wird mit dem Goldstandard verglichen
  - Kostenfunktion (loss/objective function) wird mit SGD minimiert
- Minimalisieren des Rechenaufwands durch
  - Negative Sampling
  - Subsampling

sent2vec

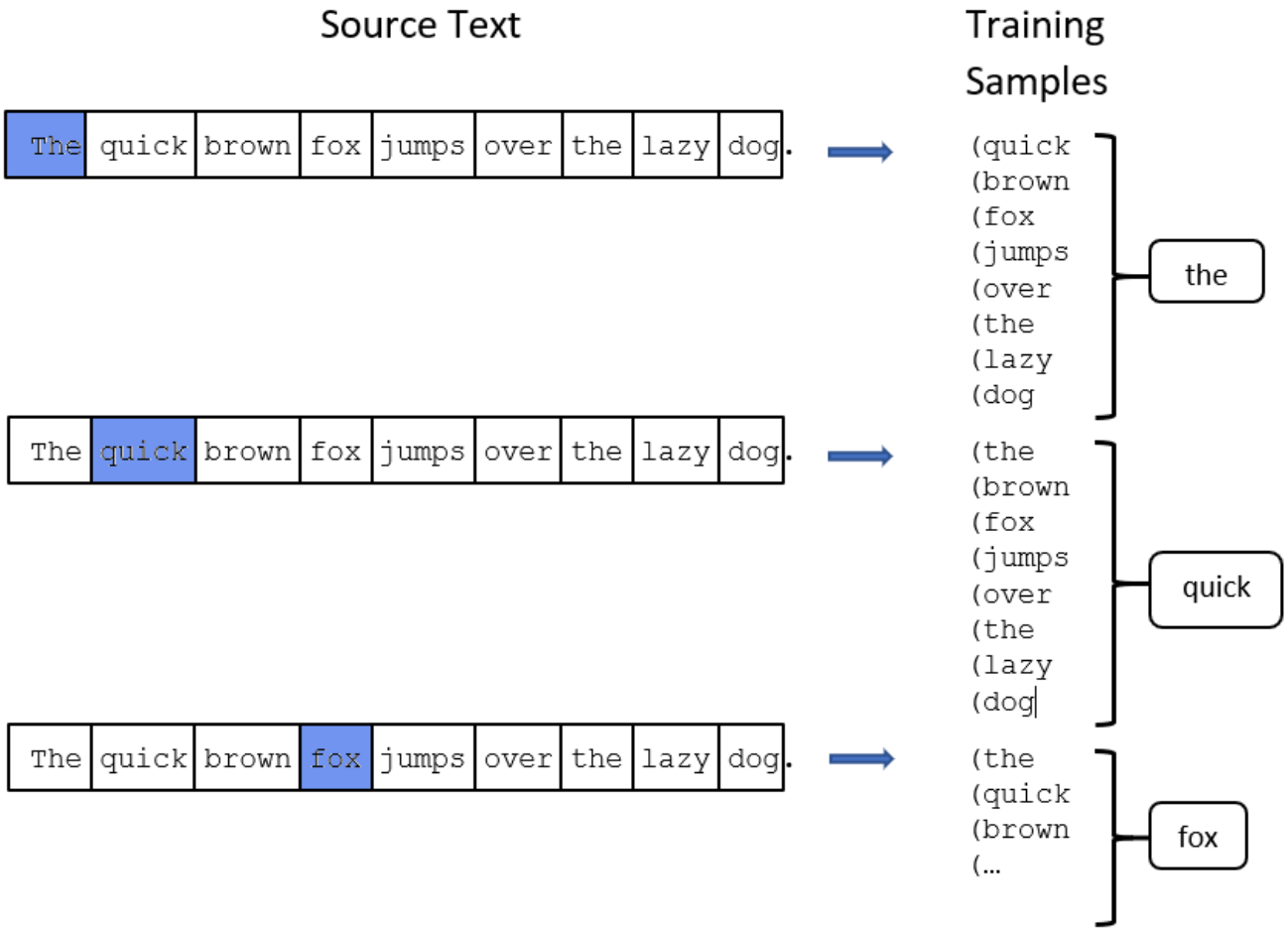
sent2vec

Was ist sent2vec?

- Erweiterung des word2vec Modells für Sätze
- Starke Anlehnung an C-BOW Modell
- Verwendet Durchschnitt von Wort- und N-Gramm-Vektoren als Satzvektor
- Reihenfolge der Sätze irrelevant für Training
- Bessere Performance (später mehr bei Evaluation)

Recap C-BOW

Gleiches Beispiel für sent2vec



sent2vec

Allgemeine Formel:

$$\begin{aligned} v_S &:= \frac{1}{R(S)} V \iota_{R(S)} \\ &= \frac{1}{R(S)} \sum_{w \in R(S)} v_w \end{aligned} \tag{2}$$

wobei

- $R(S)$ = Liste aller N-Gramme (inklusive Unigramme) im Satz S
- v_w = der entsprechende Vektor für das Kontext-Wort w
- $V \iota_{R(S)}$ ist die Summe der einzelnen Wort- und Bigramm-Vektoren, wobei V die Gewichtematrix und $\iota_{R(S)}$ die One-Hot-Vektoren der Wörter darstellt.

sent2vec

Trainingsfunktion:

$$\sum_{S \in C} \sum_{w_t \in S} \left(q_p(w_t) \ell(u_{w_t}^T v_{S \setminus \{w_t\}}) + |N_{w_t}| \sum_{w' \in \mathcal{V}} q_n(w') \ell(-u_{w'}^T v_{S \setminus \{w_t\}}) \right) \quad (3)$$

sent2vec

Trainingsfunktion:

$$\sum_{S \in C} \sum_{w_t \in S} \left(q_p(w_t) \ell(u_{w_t}^T v_{S \setminus \{w_t\}}) + |N_{w_t}| \sum_{w' \in \mathcal{V}} q_n(w') \ell(-u_{w'}^T v_{S \setminus \{w_t\}}) \right) \quad (4)$$

wobei

- $q_p(w_t)$ = Wahrscheinlichkeit $q_p(w_t)$, dass Wort w_t behalten wird
- ℓ = Kostenfunktion $\ell(x \mapsto log(1+e^{-x})$
- S = aktueller Satz S

sent2vec

Trainingsfunktion:

$$\sum_{S \in C} \sum_{w_t \in S} \left(q_p(w_t) \ell(u_{w_t}^T v_{S \setminus \{w_t\}}) + |N_{w_t}| \sum_{w' \in \mathcal{V}} q_n(w') \ell(-u_{w'}^T v_{S \setminus \{w_t\}}) \right) \quad (5)$$

wobei

- N_{w_t} = Set der negativ gesampleten Wörter $w_t \in S$
- ℓ = Kostenfunktion $\ell(x \mapsto log(1+e^{-x})$
- $q_n(w')$ = Negative-Sampling-Wahrscheinlichkeit für jedes Wort im Korpus

sent2vec & C-BOW

Unterschiede:

- Kontext immer kompletter Satz
- sent2vec benutzt zusätzlich N-gramme (hauptsächlich Bi-Gramme) als Features

Gemeinsamkeiten:

- Softmax für Output y
- Negative Sampling zur Minimalisierung des Rechenaufwands
- Subsampling für hochfrequente Wörter

sent2vec

Trainingsdaten des Modells:

- Toronto book corpus (70 Mio. Sätze, 0,9 Mrd. Wörter
- Wikipedia sentences
  Beide mithilfe der *Stanford NLP library* tokenisiert (Manning et al., 2014)

- Tweets
  Mithilfe des *NLTK tweets tokenizer* tokenisiert (Bird et al., 2009)

Evaluation

- Getestet mit 8 Testsätzen
- Verglichen mit 13 anderen Ansätzen
- Hier ein paar Beispiel Tasks

Evaluation | Supervised Testdata

- **Paraphrase Identification -> MSRP**
- **Subjectivity Classification -> SUBJ**
- **Question Type Classification -> TREC**

Microsoft Research Paraphrase Corpus | Dolan et al., 2004

- Enthält 5081 Paraphrasen
- Ursprünglich aus 9 Millioenen Sätzen eines news Korpus
- Sätze wurden mit heuristic und svm Methoden extrahiert
- Als Paraphrase / nicht Paraphrase Annotiert

Microsoft Research Paraphrase Corpus | Dolan et al., 2004

Heuristics z.B.:

- Wörter in beiden Sätzen immer: $5 \geq n \leq 40$
- Beide Sätze haben mind. 3 gleiche Worte
- Der kürzere Satz kann minimal 66.6 % des größeren sein.

Microsoft Research Paraphrase Corpus

SVM features:

- **String Similarity Features:** z.B. Wortlänge, gemeinsame Wörter
- **Morphological variants:** Morphologische Paare aus einem Lexikon
- **WordNet Lexical Mappings:** Synonym und Hyperonym Paare

Microsoft Research Paraphrase Corpus | Dolan et al., 2004

Example 1:

1) Charles O. Prince, 53, was named as Mr. Weill's successor.
2) Mr. Weill's longtime confidant, Charles O. Prince, 53, was named as his successor.

Example 2:

1) David Gest has sued his estranged wife Liza Minelli for %MONEY% million for beating him when she was drunk.
2) Liza Minelli's estranged husband is taking her to court for %MONEY% million after saying she threw a lamp at him and beat him in drunken rages.

Subjectivity Classification | Pang and Lee, 2004

- Korpus mit 10000 Sätzen
- Davon 5000 subjektiv und 5000 objektiv
- Objective Sätze von IMDb Filmzusammenfassungen
- Subjektive Sätze von Rotten Tomatoes Filmkritiken

Beispiele:
Objektiv: the movie begins in the past where a young boy named sam attempts to save celebi from a hunter .
Subjektiv: a dark , quirky road movie that constantly defies expectation .

Question Answering | Voorheers 2002

- Ein Task des TREC Korpus. (Text REtrival Conference)
- Korpus wird in Tracks eingeteilt, z.B.:
  - Web Track
  - Question Answering Track
  - News Track
- Jeder Track hat seine eigenen Tasks

Beispiel aus dem Dokument Krugman::

For which newspaper does Krugman write?
At which university does Krugman teach?

Evaluation | Unsupervised Testdata

- **Semantic Textual Similarity -> STS**
- **Sentences Involving Compositional Knowledge -> SICK 2014**

Semantic Textual Similarity | Agirre et al., 2014

- Datenset beinhaltet 3770 Satzpaare
- Task: Gib jedem Satz eine Punktzahl von 0-5:

  5 "The two sentences are completely equivalent"
  4 "The two sentences are mostly equivalent"
  3 "The two sentences are roughly equivalent"
  2 "The two sentences are not equivalent, but share details"
  1 "The two sentences are not equivalent, but share topic"
  0 "The two sentences are not equivalent"

- Daten sind aus verschiedenen Quellen:
- z.B. Tweets, News, Bildbeschreibungen

Semantic Textual Similarity | Agirre et al., 2014

Examples:

- *They flew out of the nest in groups.*
  They flew into the nest together.
- *In May 2010, the troops attempted to invade Kabul.*
  The US army invaded Kabul on May 7th last year, 2010.
- *The bird is bathing in the sink.*
  Birdie is washing itself in the water basin.

Semantic Textual Similarity | Agirre et al., 2014

Examples:

- *(2) They flew out of the nest in groups.*
  They flew into the nest together.
- *In May 2010, the troops attempted to invade Kabul.*
  The US army invaded Kabul on May 7th last year, 2010.
- *The bird is bathing in the sink.*
  Birdie is washing itself in the water basin.

Semantic Textual Similarity | Agirre et al., 2014

Examples:

- *(2) They flew out of the nest in groups.*
  They flew into the nest together.
- *(4) In May 2010, the troops attempted to invade Kabul.*
  The US army invaded Kabul on May 7th last year, 2010.
- *The bird is bathing in the sink.*
  Birdie is washing itself in the water basin.

Semantic Textual Similarity | Agirre et al., 2014

Examples:

- *(2) They flew out of the nest in groups.*
  They flew into the nest together.
- *(4) In May 2010, the troops attempted to invade Kabul.*
  The US army invaded Kabul on May 7th last year, 2010.
- *(5) The bird is bathing in the sink.*
  Birdie is washing itself in the water basin.

Sentences Involving Compositional Knowledge | Marelli et al., 2014

- Datensatz beinhaltet 10.000 Satzpaare
- Aus 1500 ausgesuchten Sätzen aufgebaut:
  - 1. Normalisierung der Sätze
  - 2. Expansion der Sätze
  - 3. Paaren der Sätze
- Annotierte Ähnlichkeitsbewertungen durch crowd sourcing
- Annotiert von 1-5, 5 ist sehr ähnlich
- Es gab insgesammt 200.000 Beurteilungen

Sentences Involving Compositional Knowledge | Marelli et al., 2014

Beispielregeln für Normalisierung:

- ***Replace Named Entities with a word that stands for the class.***
- *S0: "A woman is playing Mozart"*
  S1: "A woman is playing classical music"

Beispielregeln für Expansion:

- ***Turn active sentences into passive sentences and viceversa.***
- *S1: "A man is driving a car "*
  S2: "The car is being driven by a man"
- ***Insert or remove negations to produce contradictions.***
- *S1: "The boy is playing the piano"*
  S3: "The boy is not playing the piano"

Evaluation | Vergleich

Vergleich auf supervised Tasks:

Data	Model	MSRP (Acc / F1)	MR	CR	SUBJ	MPQA	TREC	Average
Unordered Sentences: (Toronto Books; 70 million sentences, 0.9 Billion Words)	SAE	**74.3** / 81.7	62.6	68.0	86.1	76.8	80.2	74.7
	SAE + embs.	70.6 / 77.9	73.2	75.3	89.8	86.2	80.4	79.3
	SDAE	**76.4 / 83.4**	67.6	74.0	89.3	81.3	77.7	78.3
	SDAE + embs.	**73.7** / 80.7	74.6	78.0	90.8	**86.9**	78.4	80.4
	ParagraphVec DBOW	72.9 / 81.1	60.2	66.9	76.3	70.7	59.4	67.7
	ParagraphVec DM	73.6 / **81.9**	61.5	68.6	76.4	78.1	55.8	69.0
	Skipgram	69.3 / 77.2	73.6	77.3	89.2	85.0	82.2	78.5
	C-BOW	67.6 / 76.1	73.6	77.3	89.1	85.0	82.2	79.1
	Unigram TFIDF	73.6 / 81.7	73.7	79.2	90.3	82.4	**85.0**	80.7
	Sent2Vec uni.	72.2 / 80.3	75.1	**80.2**	90.6	**86.3**	83.8	**81.4**
	Sent2Vec uni. + bi.	72.5 / 80.8	**75.8**	**80.3**	**91.2**	85.9	**86.4**	**82.0**
Ordered Sentences: Toronto Books	SkipThought	73.0 / **82.0**	**76.5**	**80.1**	**93.6**	**87.1**	**92.2**	**83.8**
	FastSent	72.2 / 80.3	70.8	78.4	88.7	80.6	76.8	77.9
	FastSent+AE	71.2 / 79.1	71.8	76.7	88.8	81.5	80.4	78.4
2.8 Billion words	C-PHRASE	72.2 / 79.6	**75.7**	78.8	**91.1**	86.2	78.8	80.5

Evaluation | Vergleich

Vergleich auf unsupervised Tasks:

Model	STS 2014 News	Forum	WordNet	Twitter	Images	Headlines	SICK 2014 Test + Train	Average
SAE	.17/.16	.12/.12	.30/.23	.28/.22	.49/.46	.13/.11	.32/.31	.26/.23
SAE + embs.	.52/.54	.22/.23	.60/.55	.60/.60	.64/.64	.41/.41	.47/.49	.50/.49
SDAE	.07/.04	.11/.13	.33/.24	.44/.42	.44/.38	.36/.36	.46/.46	.31/.29
SDAE + embs.	.51/.54	.29/.29	.56/.50	.57/.58	.59/.59	.43/.44	.46/.46	.49/.49
ParagraphVec DBOW	.31/.34	.32/.32	.53/.50	.43/.46	.46/.44	.39/.41	.42/.46	.41/.42
ParagraphVec DM	.42/.46	.33/.34	.51/.48	.54/.57	.32/.30	.46/.47	.44/.40	.43/.43
Skipgram	.56/.59	.42/.42	.73/**.70**	**.71**/.74	.65/.67	.55/.58	.60/.69	.60/.63
C-BOW	.57/.61	**.43/.44**	.72/.69	**.71/.75**	.71/.73	.55/.59	.60/.69	.60/.65
Unigram TF-IDF	.48/.48	.40/.38	.60/.59	.63/.65	.72/.74	.49/.49	.52/.58	.55/.56
Sent2Vec uni.	**.62/.67**	**.49/.49**	**.75/.72**	.70/**.75**	**.78/.82**	**.61/.63**	**.61/.70**	**.65/.68**
Sent2Vec uni. + bi.	**.62/.67**	**.51/.51**	.71/.68	.70/**.75**	**.75/.79**	.59/.62	**.62/.70**	**.65/.67**
SkipThought	.44/.45	.14/.15	.39/.34	.42/.43	.55/.60	.43/.44	.57/.60	.42/.43
FastSent	.58/.59	.41/.36	**.74/.70**	.63/.66	.74/.78	.57/.59	**.61/.72**	.61/.63
FastSent+AE	.56/.59	.41/.40	.69/.64	.70/.74	.63/.65	.58/.60	.60/.65	.60/.61
Siamese C-BOW[4]	.58/.59	.42/.41	.66/.61	**.71**/.73	.65/.65	**.63/.64**	–	–
C-PHRASE	**.69/.71**	**.43**/.41	**.76/.73**	.60/.65	**.75/.79**	**.60/.65**	.60/**.72**	**.63/.67**

Evaluation | Vergleich

Vergleich der Makroähnlichkeiten von unsupervised Algorithmen:

Type	Training corpus	Method	Supervised average	Unsupervised average	Macro average	Training time (in hours)
unsupervised	twitter (19.7B words)	**Sent2Vec uni. + bi.**	83.5	68.3	75.9	6.5*
unsupervised	twitter (19.7B words)	**Sent2Vec uni.**	82.2	69.0	75.6	3*
unsupervised	Wikipedia (1.7B words)	**Sent2Vec uni. + bi.**	83.3	66.2	74.8	2*
unsupervised	Wikipedia (1.7B words)	**Sent2Vec uni.**	82.4	66.3	74.3	3.5*
unsupervised	Toronto books (0.9B words)	**Sent2Vec books uni.**	81.4	66.7	74.0	1*
unsupervised	Toronto books (0.9B words)	**Sent2Vec books uni. + bi.**	82.0	65.9	74.0	1.2*
semi-supervised	structured dictionary dataset	DictRep BOW + emb	80.5	66.9	73.7	24**
unsupervised	2.8B words + parse info.	C-PHRASE	80.5	64.9	72.7	–
unsupervised	Toronto books (0.9B words)	C-BOW	79.1	62.8	70.2	2
unsupervised	Toronto books (0.9B words)	FastSent	77.9	62.0	70.0	2
unsupervised	Toronto books (0.9B words)	SkipThought	83.8	42.5	63.1	336**

Conclusion | Discussion

- Starke Generalisierbarkeit, robust auf vielen Tasks
- Schnell zu trainieren, nur etwas langsamer als CBOW
- Bester Algorithmus auf Makrodurchschnitt
- Könnte durch vortrainierte Embeddings verbessert werden

References

- Steven Bird, Ewan Klein, Edward Loper (2009): Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc."
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, David McClosky (2014): The stanford corenlp natural language processing toolkit. In ACL (System Demonstrations). pages 55–60
- Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean (2013) : Efficient Estimation of Word Representations in Vector Space, https://arxiv.org/pdf/1301.3781.pdf
- Matteo Pagliardini, Prakhar Gupta, Martin Jaggi (2017, last updated 2018) : Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features, https://arxiv.org/pdf/1703.02507.pdf
- Xin Rong (2016): word2vec Parameter Learning Explained, https://arxiv.org/pdf/1411.2738.pdf

References

- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014). Association for Computational Linguistics Dublin, Ireland, pages 81–91.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In LREC. pages 216–223.
- Ellen M Voorhees. 2002. Overview of the trec 2001 question answering track. In NIST special publication. pages 42–51.