word2vec

Durgesh Kumar
OSINT LAB, CSE Department
IIT Guwahati

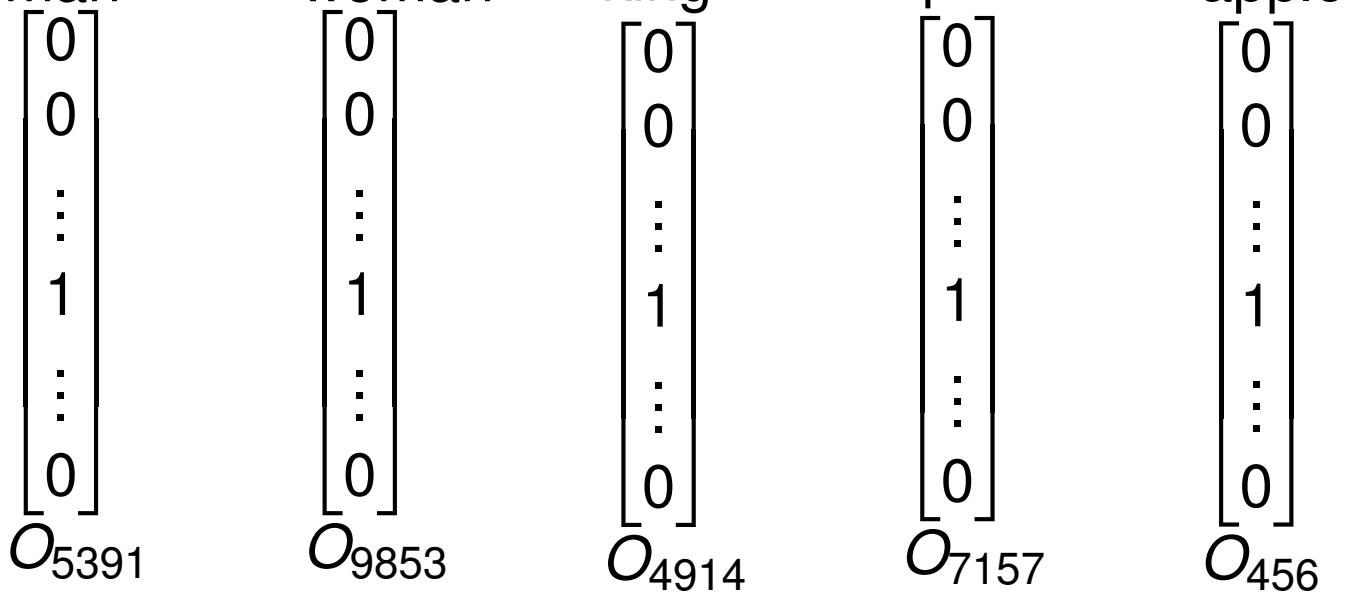
Table of contents

Word Representation

One hot vector

- V= [a, aaron, ..., apple, ..., man, ..., woman, ..., king, ..., queen, ..., zula, <UNK>]

man O_{5391} woman O_{9853} king O_{4914} queen O_{7157} apple O_{456}



Weakness

- It treats each word as discrete thing
- it does not allow to utilize the inter word relationship
- I want a glass of orange —— .

[1]

[1] This slide is borrowed from the lecture of deeplearning.ai

Featurized representation : word embedding

	a	*aaron*	⋯	*apple*	*man*	*woman*	*king*	*queen*	*orange*
Gender	0.00	0.004	⋯	−1	1	−0.95	0.97	−21	46
Royal	0.01	0.02	⋯	0.28	0.21	0.11	11	1.46	0.12
Age	0.03	0.02	⋯	−0.36	0.84	0.13	5.68	13	2.19
Food	0.09	0.01	⋯	15	1.67	1.14	0.09	2.2	12
noun	2.3	−2.4	⋯	0.01	0.05	8.10	1.4	1.2	1.6
verb	2.3	−2.4	⋯	0.01	0.05	8.10	1.4	1.2	1.6

Introduction to word2vec

- **word2vec** is one of the popular model to learn word embedding
- **word embedding** is dense vector of fixed size representing a word capturing semantic and syntactic regularities
  - semantic regularities : Antonym, synonym, etc
  - syntactic regularities : language structure, verb, noun, etc
  - each word is represented by a vector of fixed dimension varying from 50 to 300.
  - boy : [0.89461, 0.37758, 0.42067, -0.51334, -0.28298, 1.0012, 0.18748, 0.21868, -0.030053, ...]
- word2vec is proposed by T. Milkov et. al. in 2013.
  - **paper [1]**: Efficient estimation of word representations in vector space by T. Mikolov et. al. in *ICLR Workshop* 2013
  - **paper [2]**: Distributed Representations of Words and Phrases and their Compositionality by T. Mikolov et. al. in *NIPS* 2013

Interesting examples of semantic and syntactic relations

Examples from **paper [1]**: Efficient estimation of word representations in vector space by T. Mikolov et. al. in *ICLR Workshop* 2013

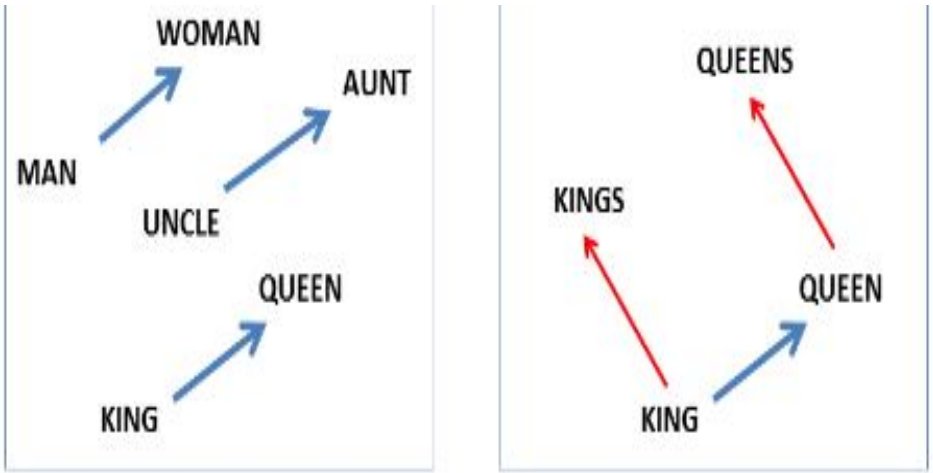
- vector(“king”) - vector(“man”) + vector(“woman”) is closest to vector(“queen”)
- vector(“big”) : vector(“biggest”) : : vector(“small”) : vector(“smallest”)

Interesting examples of semantic and syntactic relations

Examples from **paper [1]**: Efficient estimation of word representations in vector space by T. Mikolov et. al. in *ICLR Workshop* 2013

- vector(“king”) - vector(“man”) + vector(“woman”) is closest to vector(“queen”)
- vector(“big”) : vector(“biggest”) : : vector(“small”) : vector(“smallest”)

Figure: Word pairs illustrating the gender relation and singular/plural relations from **paper [3]**



More examples from Paper [1]

Table: Examples of five types of semantic and nine types of syntactic word relationship

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

More examples from Paper [1]

Table: Examples of the word pair relationships, using the best word vectors (Skip-gram model trained on 783M words with 300 dimensionality)

Relationship	**Example 1**	**Example 2**	**Example 3**
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Few terminologies related to the word2vec model

Target word, context word , sliding window

- The yellow quick brown fox jumps over the lazy dog
  - *target word*: fox
  - *context word*: quick, brown, jumps, over
  - *window length*: 5

Few terminologies related to the word2vec model

Target word, context word , sliding window

- The yellow quick brown fox jumps over the lazy dog
  - *target word*: fox
  - *context word*: quick, brown, jumps, over
  - *window length*: 5
- The yellow quick brown fox jumps over the lazy dog

Few terminologies related to the word2vec model

Target word, context word , sliding window

- The yellow quick brown fox jumps over the lazy dog
  - *target word*: fox
  - *context word*: quick, brown, jumps, over
  - *window length*: 5
- The yellow quick brown fox jumps over the lazy dog
- The yellow quick brown fox jumps over the lazy dog

One hot vector encoding and Embedding matrix

- The yellow quick brown fox jumps over the lazy dog
- Let V = {the, yellow, quick, brown, fox, jumps, over, lazy, dog } ; Ordered dictionary of unique word in the corpus
- 9 unique word in the corpus
- the : $[1, 0, 0, 0, 0, 0, 0, 0, 0]^T \rightarrow O_1$
- yellow : $[0, 1, 0, 0, 0, 0, 0, 0, 0]^T \rightarrow O_2$
- brown : $[0, 0, 0, 1, 0, 0, 0, 0, 0]^T \rightarrow O_4$

Embedding Matrix

$$E_{5*9} = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{array} \begin{array}{c} \begin{array}{ccccccccc} \textit{the} & \textit{yellow} & \textit{quick} & \textit{brown} & \textit{fox} & \textit{jumps} & \textit{over} & \textit{lazy} & \textit{dog} \end{array} \\ \begin{bmatrix} -1 & 1 & 0.04 & 1.24 & 1.12 & 1.21 & 4 & -21 & 46 \\ 0.01 & 0.02 & 1.56 & 0.28 & 0.21 & 0.11 & 11 & 1.46 & 61 \\ 0.03 & 0.02 & -0.36 & 0.84 & 0.13 & 5.68 & 13 & 2.19 & 72 \\ 0.09 & 0.01 & 0.09 & 1.67 & 1.14 & 0.09 & 2.2 & 3.8 & 49 \\ 2.3 & -2.4 & 0.01 & 0.05 & 8.10 & 1.4 & 1.2 & 1.6 & 1.8 \end{bmatrix} \end{array}$$

- $O_1 = [\ 1, 0, 0\ , 0, 0, 0, 0, 0, 0]^T$
- $E_{5*9}\ .\ O_{1_{(9*1)}} = e_{1_{(5*1)}} \rightarrow [\ \text{-1}, 0.01, 0.03, 0.09\ , 2.3\ ]^T$

CBOW and skipgram architecture

The yellow quick brown fox jumps over the lazy dog

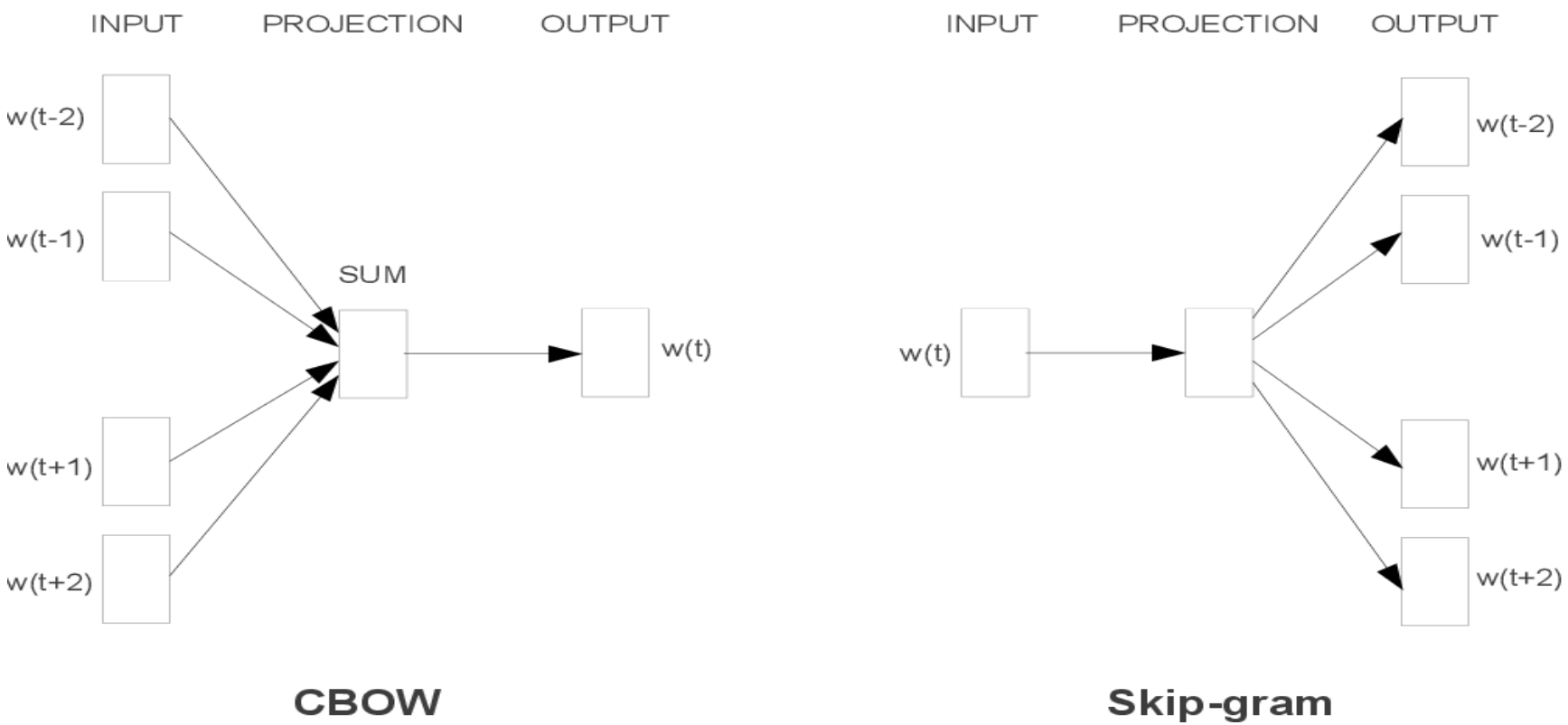


Figure: The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word [1]

CBOW and skipgram architecture

The yellow quick brown fox jumps over the lazy dog

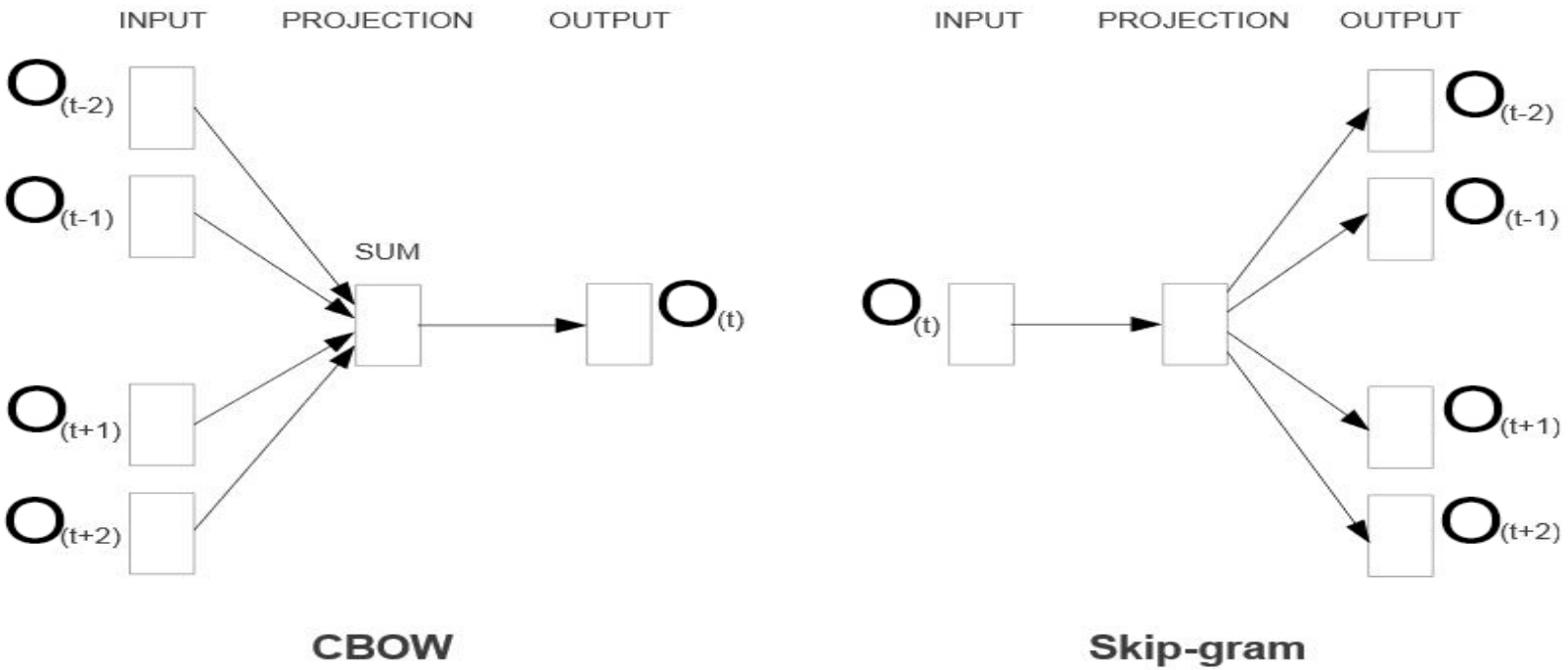
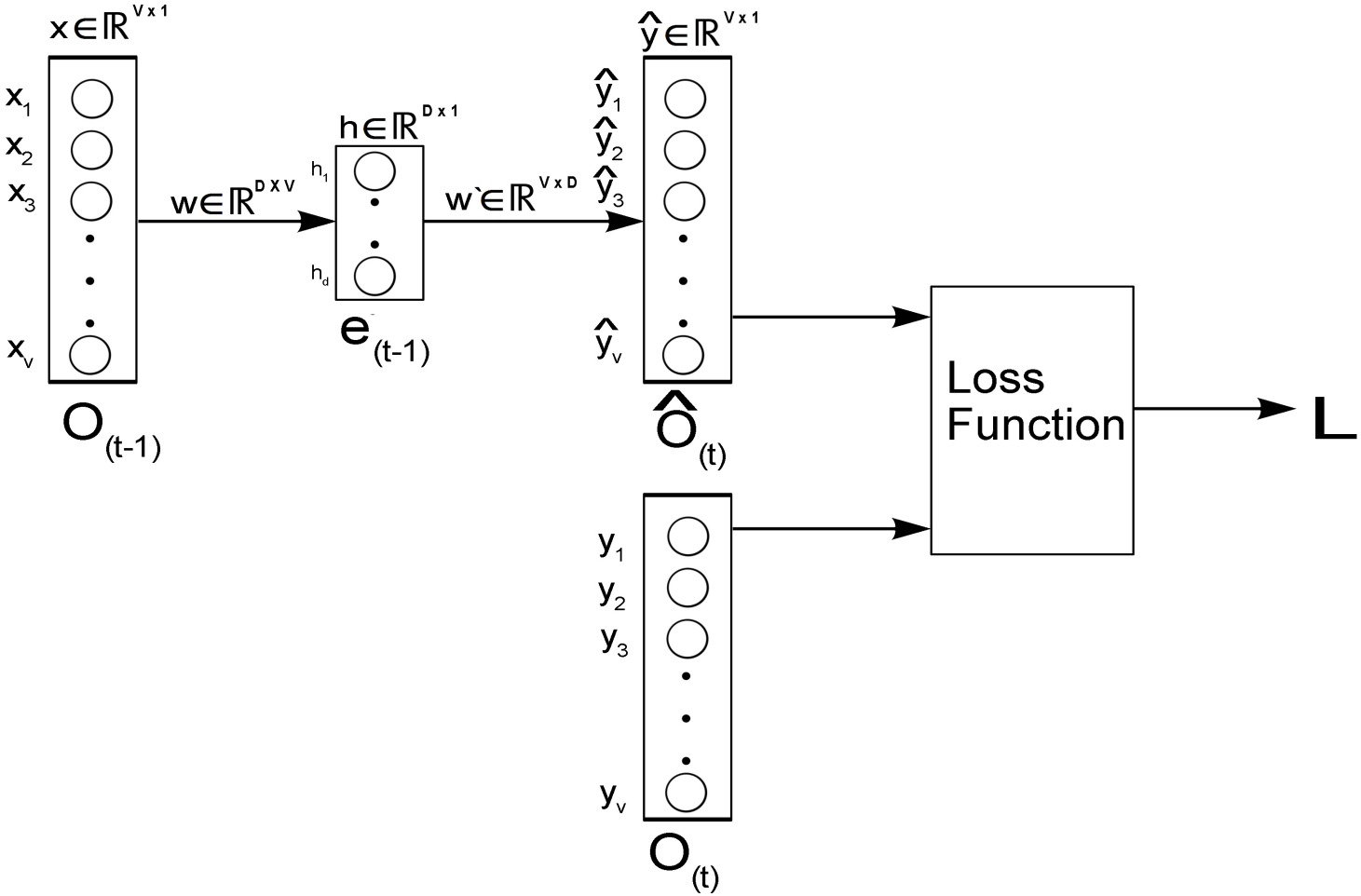


Figure: The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word [1]

CBOW simplified architecture

The yellow quick brown fox jumps over the lazy dog



CBOW vs Skipgram

- Skipgram is better at predicting syntactic relationship
- CBOW is aprox 20 times faster than skipgram
- Both CBOW and skipgram are good at predicting semantic relationship

References I

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
Distributed representations of words and phrases and their compositionality.
In *Advances in neural information processing systems*, pages 3111–3119, 2013.

References II

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig.
Linguistic regularities in continuous space word representations.
In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.