Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Kai Chen
Greg Corrado
Jeffrey Dean

Wonpyo Hong
2021311625

Contents

Terminology

- N-gram
- Skip-gram
- NNLM
- RNNLM
- LSA
- LDA
- Hierarchical Softmax
- Adagrad
- Bag-of-Words

Research Goal

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

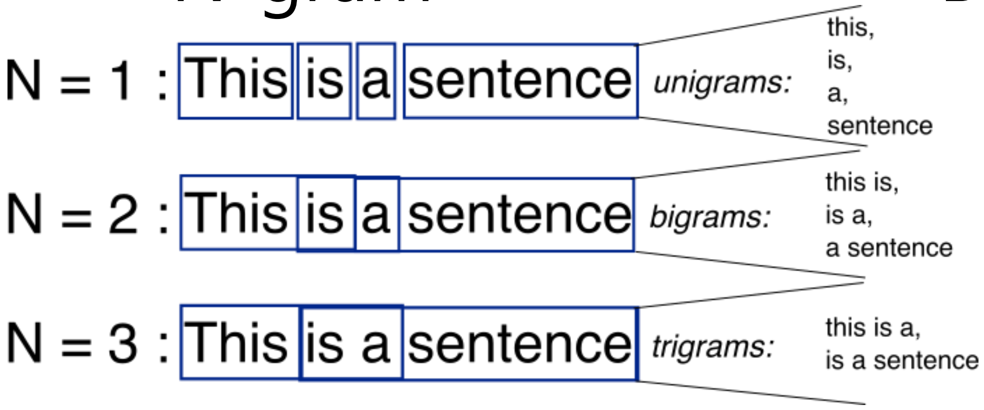
- Vector Representations of Words
  - Two Novel Model Architectures
  - Improving Accuracy While Lowering Cost

Key Related Work

- Representation of Texts
  - Local Representations
    - N-grams
    - Bag-of-Words
    - One-hot vector
  - Continuous Representations
    - LSA
    - LDA
    - Distributed Representations
- NNLM
  - Feedforward Neural Net Language Model
- RNNLM
  - Recurrent Neural Net language Model

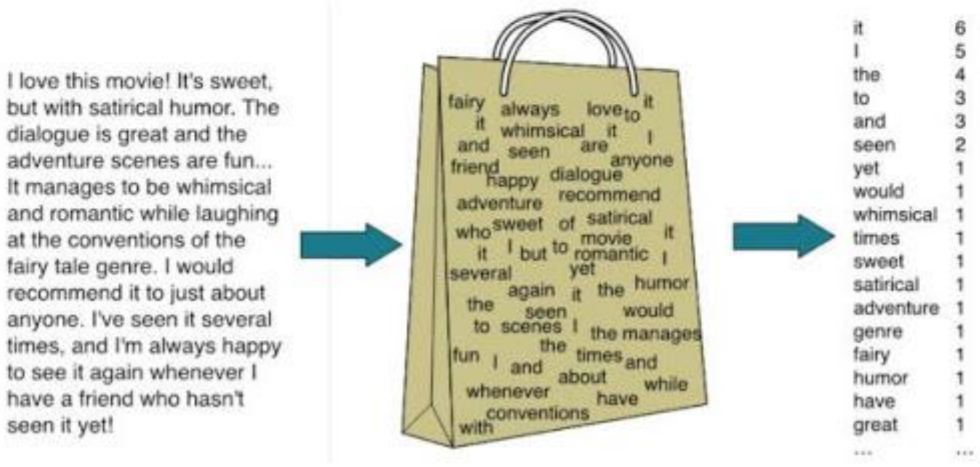
Local Representations

N-gram

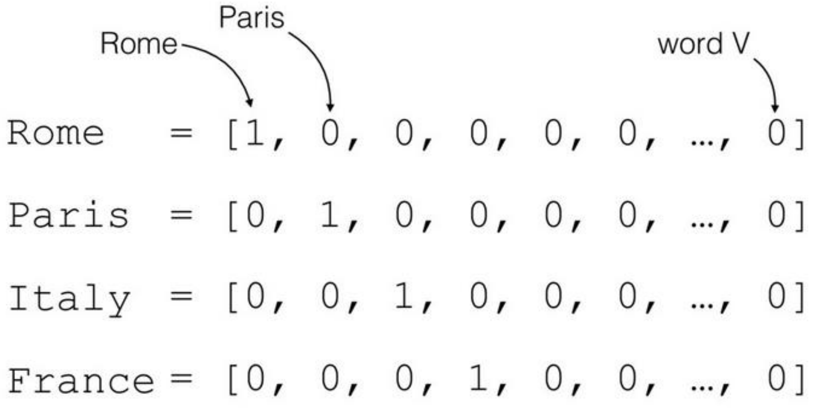


https://stackoverflow.com/questions/18193253/what-exactly-is-an-n-gram

Bag-of-Words



One-hot vector



(Marco Bonzanini, 2017)

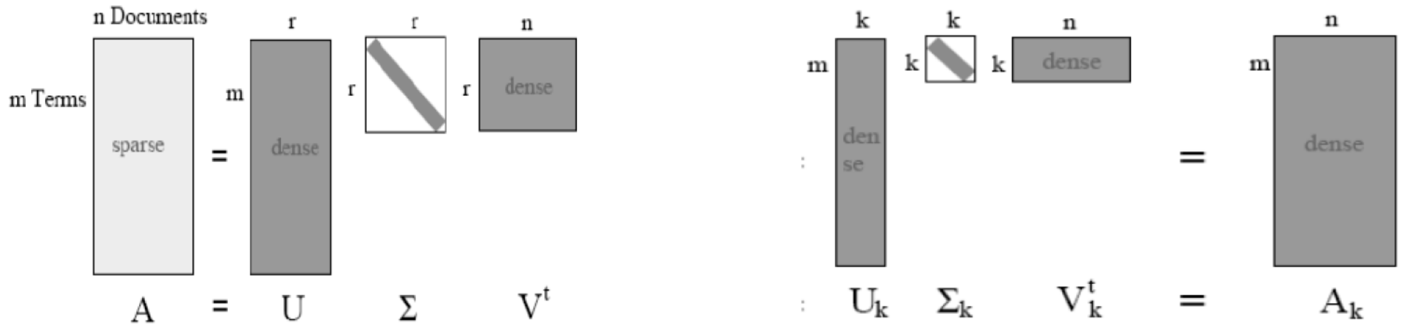
LSA & LDA

- LSA
  - Latent Semantic Analysis
- LDA
  - Latent Dirichlet Allocation

- Topic Modeling
  - Discover abstract topic that occurs in a collection of documents

LSA & LDA

- SVD (truncated SVD)



- LSA focuses on reducing matrix dimension
  - Small scale only (around 100 patents)
- LDA focuses on solving the topic modeling problem

https://towardsdatascience.com/2-latent-methods-for-dimension-reduction-and-topic-modeling-20ff6d7d547
https://wikidocs.net/30708
https://heehehe-ds.tistory.com/entry/NLP-%ED%86%A0%ED%94%BD-%EB%AA%A8%EB%8D%B8%EB%A7%81-Topic-Modeling-LSA-LDA
https://par.nsf.gov/servlets/purl/10055536

- A: Cute kitty
- B: Eat rice or cake
- C: Kitty and hamster
- D: Eat bread
- E: Rice, bread and cake
- F: Cute hamster eats bread and cake

	cute	kitty	eat	rice	cake	hamster	bread
A	1	1	0	0	0	0	0
B	0	0	1	1	1	0	0
C	0	1	0	0	0	1	0
D	0	0	1	0	0	0	1
E	0	0	0	1	1	0	1
F	1	0	1	0	1	1	1

$$X = U\Sigma V^T$$

0.12	0.57	-0.32	0.00	-0.71	-0.24
0.44	-0.36	-0.41	0.71	0.00	-0.08
0.12	0.57	-0.32	0.00	0.71	-0.24
0.33	-0.07	0.56	0.00	0.00	-0.75
0.44	-0.36	-0.41	-0.71	0.00	-0.08
0.69	0.30	0.37	0.00	0.00	0.55

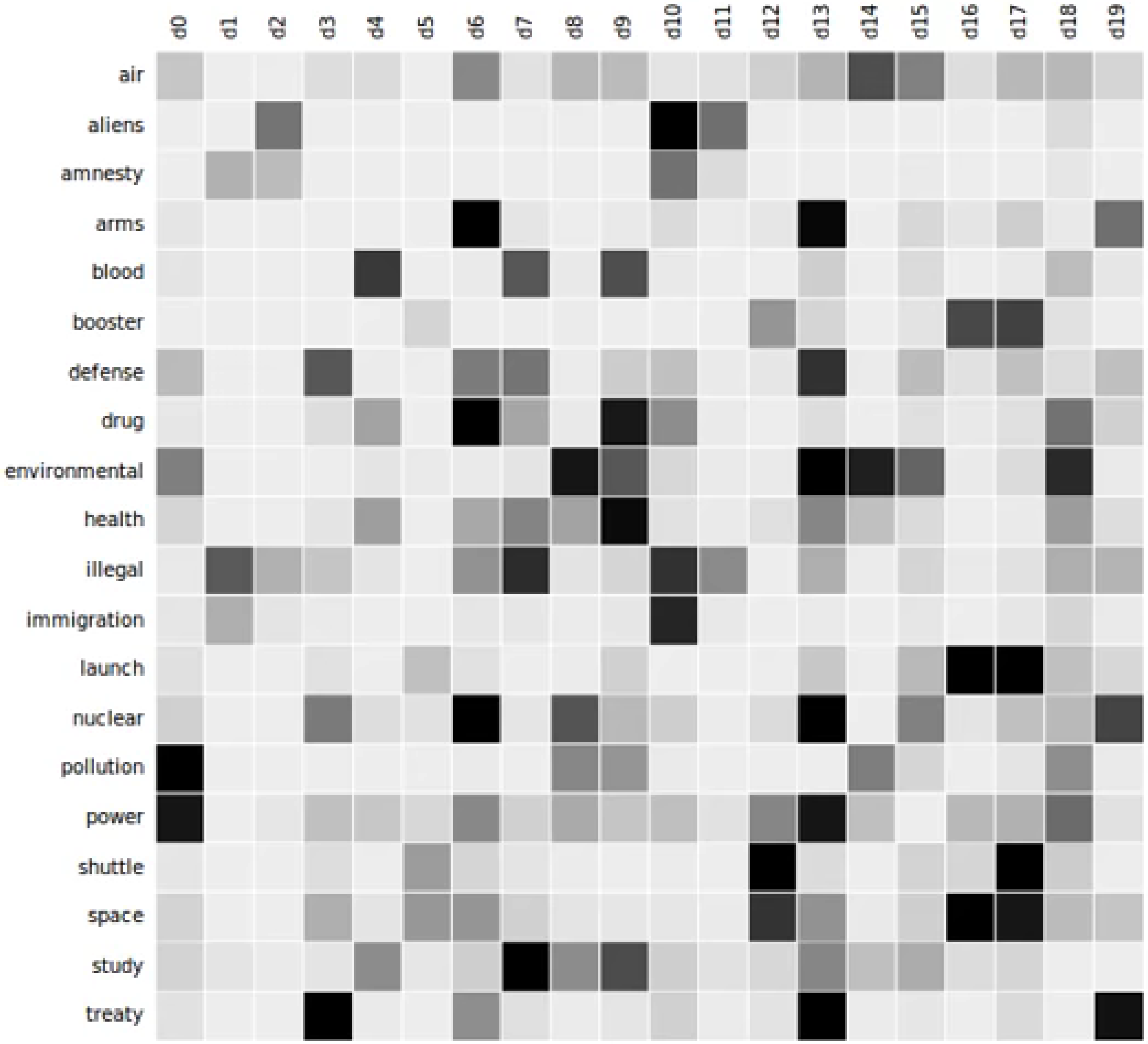
2.98	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.88	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.36	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.87	0.00

0.27	0.08	0.49	0.30	0.53	0.27	0.49
0.46	0.61	-0.07	-0.38	-0.22	0.46	-0.07
0.04	-0.47	0.38	-0.61	-0.34	0.04	0.38
0.00	0.00	0.71	0.00	0.00	0.00	-0.71
-0.71	0.00	0.00	0.00	0.00	0.71	0.00
0.35	-0.56	-0.33	-0.18	0.44	0.35	-0.33
0.30	-0.30	0.00	0.60	-0.60	0.30	0.00

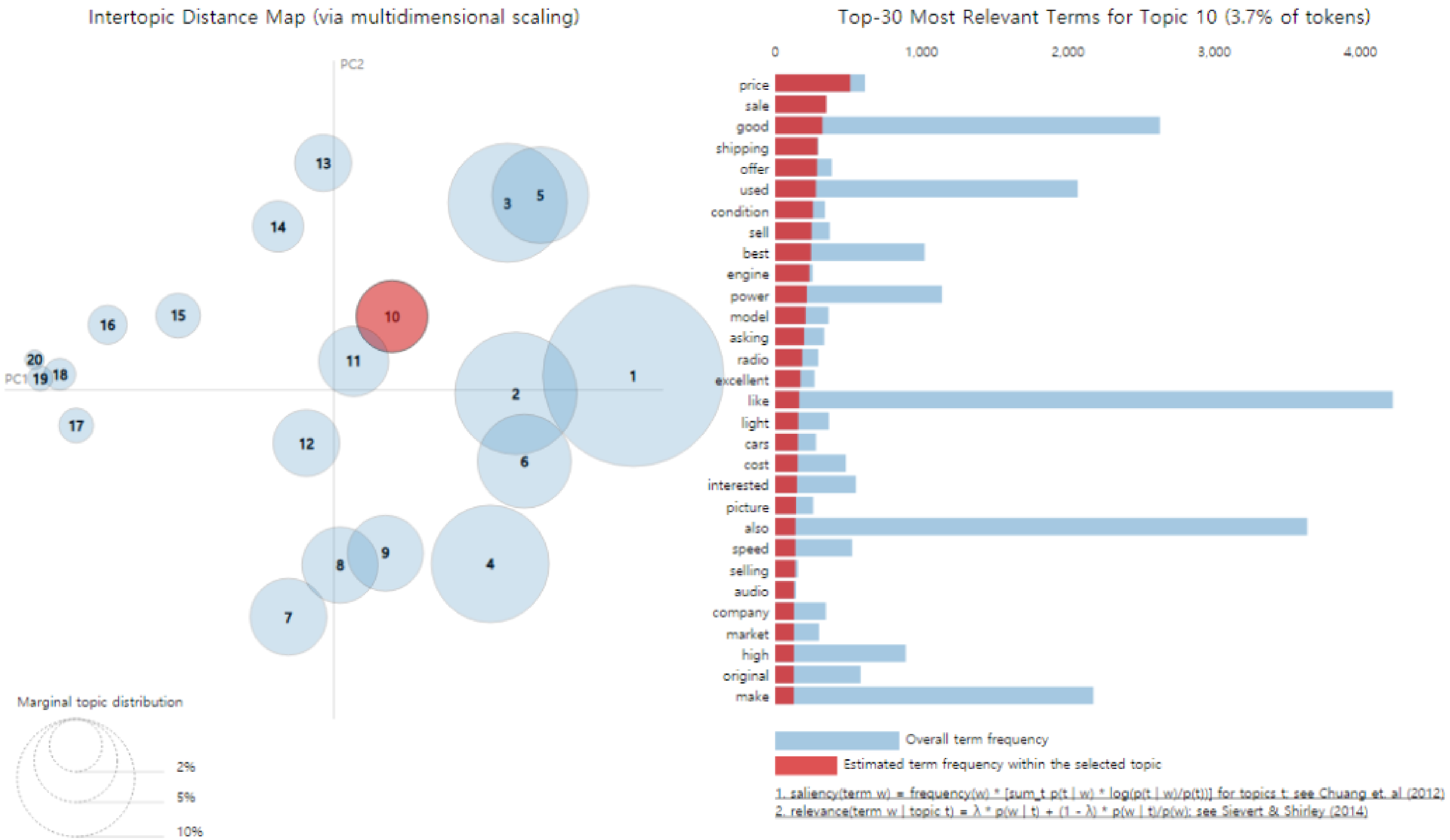
$$X_k = U_k \Sigma_k V_k^T$$

0.59	**0.68**	0.10	-0.30	-0.05	0.59	0.10
0.04	-0.31	**0.69**	**0.65**	**0.84**	0.04	0.69
0.59	**0.68**	0.10	-0.30	-0.05	**0.59**	0.10
0.20	0.00	**0.49**	0.34	0.55	0.20	**0.49**
0.04	-0.31	0.69	**0.65**	**0.84**	0.04	**0.69**
0.81	0.51	**0.97**	0.40	**0.96**	**0.81**	**0.97**

Values provided from
https://bab2min.tistory.com/569


d0
d1
d2
d3
d4
d5
d6
d7
d8
d9
d10
d11
d12
d13
d14
d15
d16
d17
d18
d19
air
aliens
amnesty
arms
blood
booster
defense
drug
environmental
health
illegal
immigration
launch
nuclear
pollution
power
shuttle
space
study
treaty

LSA & LDA


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 10 (3.7% of tokens)
0
1,000
2,000
3,000
4,000
price
sale
good
shipping
offer
used
condition
sell
best
engine
power
model
asking
radio
excellent
like
light
cars
cost
interested
picture
also
speed
selling
audio
company
market
high
original
make
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

LSA & LDA

Many different types of models were proposed for estimating continuous representations of words, including the well-known Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). In this paper, we focus on distributed representations of words learned by neural networks, as it was previously shown that they perform significantly better than LSA for preserving linear regularities among words [20, 31]; LDA moreover becomes computationally very expensive on large data sets.

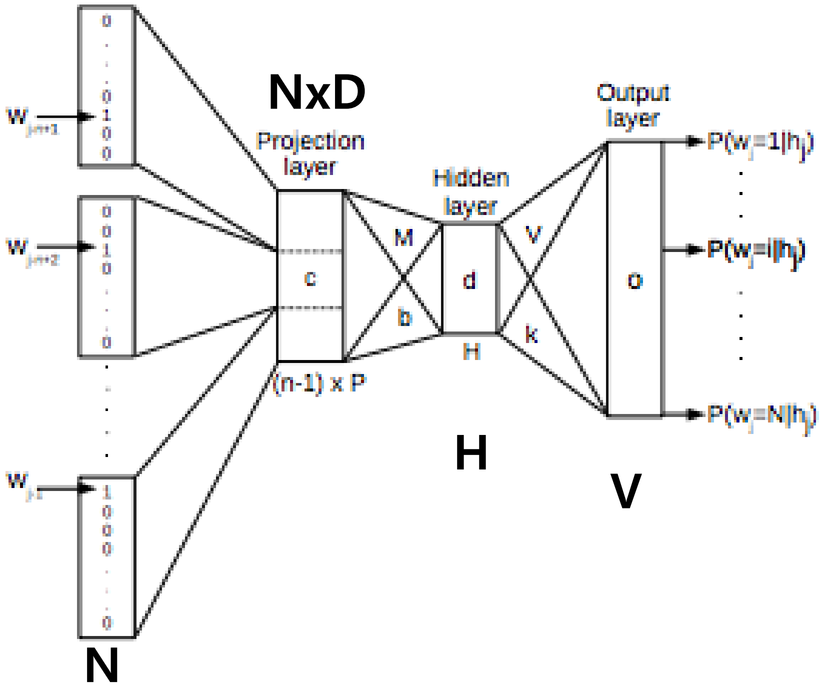
pg.2

- Linear regularities
  - Linear additive properties from vectorized form of words
  - vector("King") – vector("Man") + vector("Woman") → vector("Queen")

- Updating new words takes too much effort

NNLM (Feedforward NNLM)

- Layers
  - Input, Projection, Hidden, Output



NNLM

$$Q = N \times D + N \times D \times H + H \times V,$$

- Q = computational cost (complexity)
- N = number of previous words used for learning
- D = dimensionality of projection layer
- H = size of hidden layer
- V = size of the vocabulary and output layer

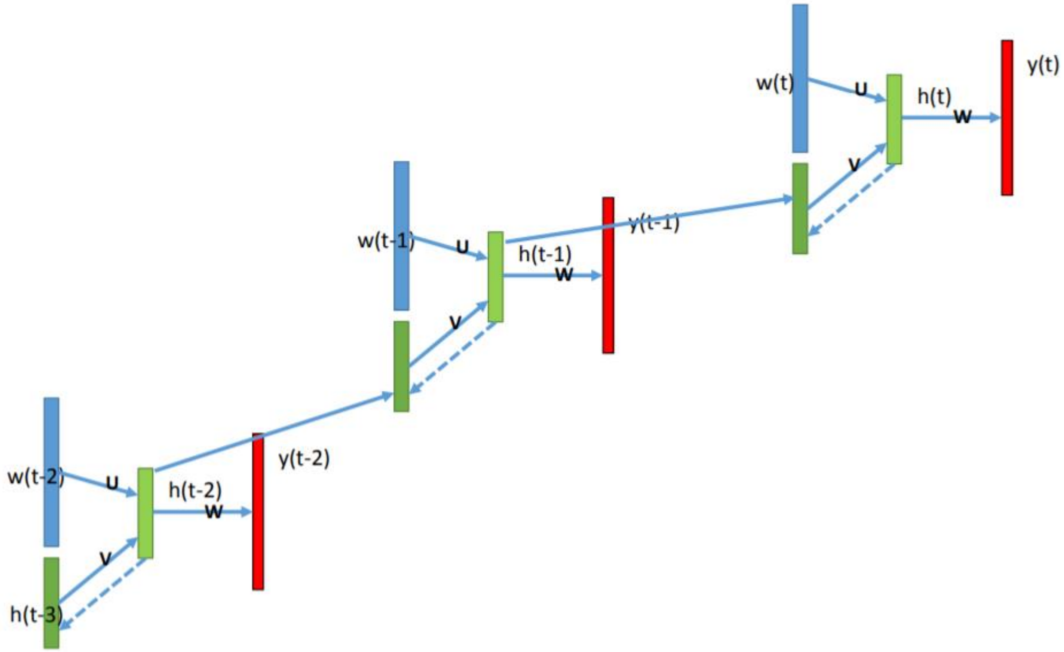
P = N x D → around 500 ~ 2000

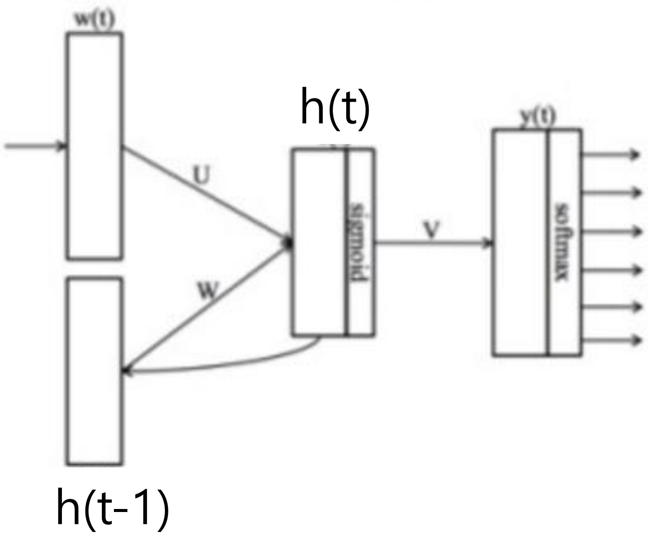
N → around 10

H → 500 ~ 1000

RNNLM (Recurrent NNLM)

- Layers
  - Input, Hidden, Output





$$e_t = lookup(x_t)$$

$$h_t = tanh(W_x e_t + W_h h_{t-1} + b)$$

$$\hat{y}_t = softmax(W_y h_t + b)$$

RNNLM

$$Q = H \times H + H \times V,$$

- Q = computational cost (complexity)
- ~~N = number of previous words used for learning~~
- ~~D = dimensionality of projection layer~~
- H = size of hidden layer
- V = size of the vocabulary and output layer

Proposed Method

- CBOW
- Continuous Skip-gram

$$Q = N \times D + D \times log_2(V).$$

$$Q = C \times (D + D \times log_2(V)),$$

- NNLM
- RNNLM

$$log_2(V)$$

$$Q = N \times D + \boxed{N \times D \times H} + H \times \cancel{V},$$

$$Q = \boxed{H \times H} + H \times \cancel{V},$$

$$log_2(V)$$

CBOW (Continuous Bag-of-Words)

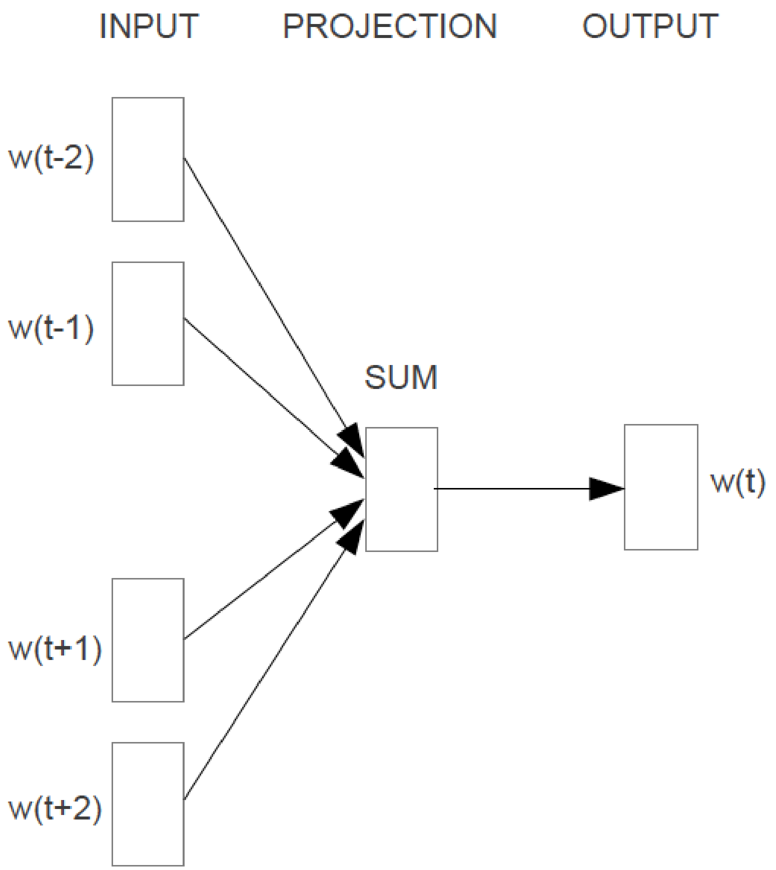
$$Q = N \times D + D \times log_2(V).$$

3.1 Continuous Bag-of-Words Model

The first proposed architecture is similar to the feedforward NNLM, where the non-linear hidden layer is removed and the projection layer is shared for all words (not just the projection matrix); thus, all words get projected into the same position (their vectors are averaged). We call this architecture a bag-of-words model as the order of words in the history does not influence the projection. Furthermore, we also use words from the future; we have obtained the best performance on the task introduced in the next section by building a log-linear classifier with four future and four history words at the input, where the training criterion is to correctly classify the current (middle) word. Training complexity is then

$$Q = N \times D + D \times log_2(V). \tag{4}$$

We denote this model further as CBOW, as unlike standard bag-of-words model, it uses continuous distributed representation of the context. The model architecture is shown at Figure 1. Note that the weight matrix between the input and the projection layer is shared for all word positions in the same way as in the NNLM.

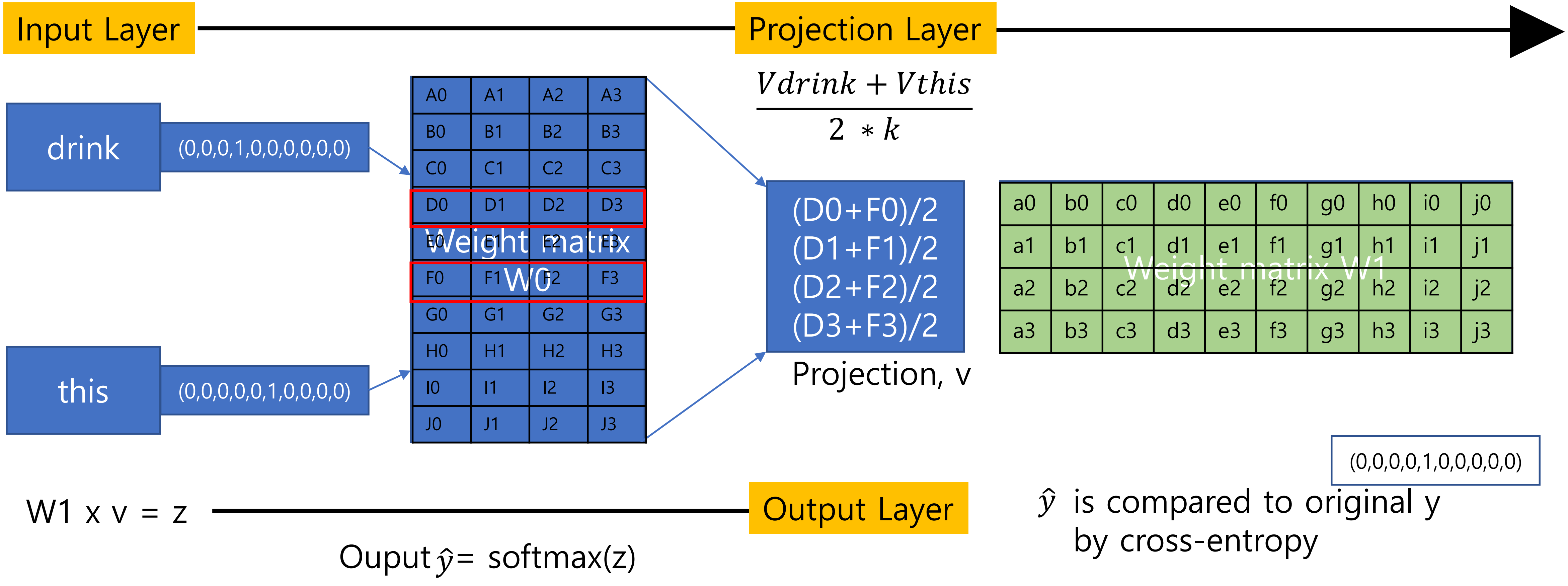


CBOW

CBOW (Continuous Bag-of-Words)

0 1 2 3 4 5 6 7 8 9

- Yet again I drink coffee this morning to study A.I.



CBOW

- Computation Cost

NNLM

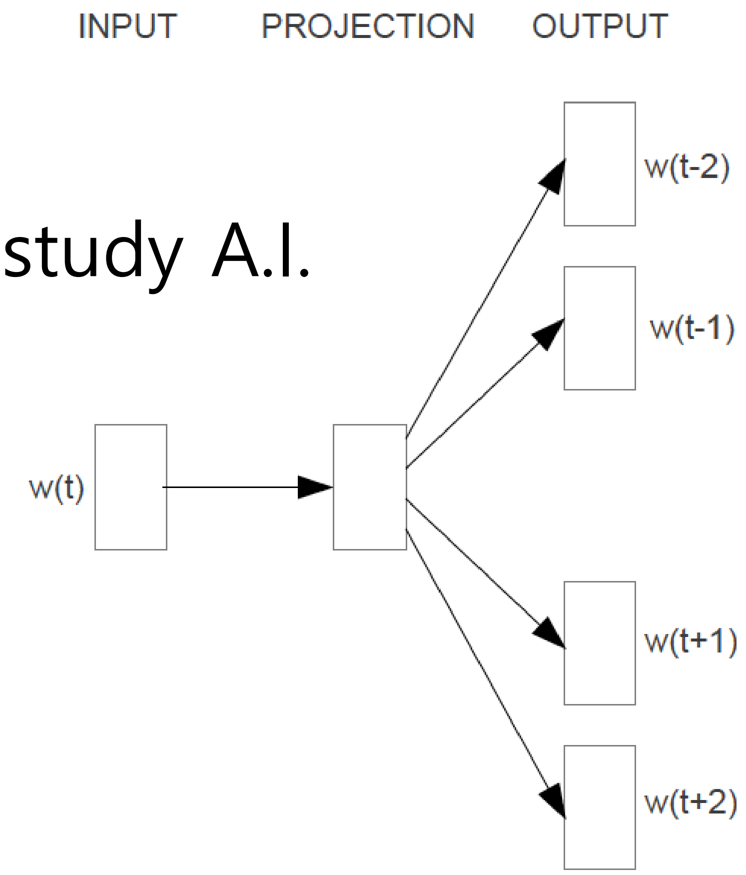
$$Q = N \times D + \cancel{\boxed{N \times D \times H}} + \underset{D}{\cancel{H}} \times \overset{log_2(V)}{\cancel{V}},$$

$$Q = N \times D + D \times log_2(V).$$

Continuous Skip-Gram

- CBOW the other way around.
- Yet again I drink $\boxed{\text{coffee}}$ this morning to study A.I.

Drink, caffeine, black, Starbucks, water, ice,
americano, barista, drip, night, sleep …



Skip-gram

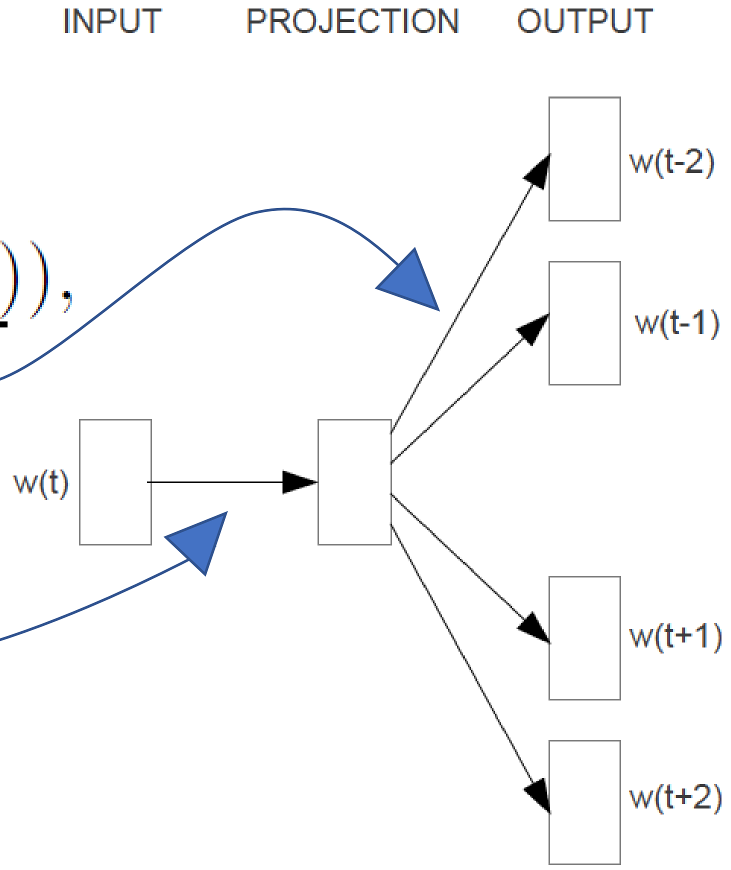
Continuous Skip-gram

- Computational Cost

$$Q = \underline{C} \times (\underline{D} + \underline{D \times log_2(V)}),$$

Softmax has to be done
for each surrounding words
to reach output layer

  - C = size of 'surrounding words'



Skip-gram

Results

- Linear Regularities

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

Results

Dimensionality / Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

$$Q = N \times D + D \times log_2(V).$$

Table 2: *Accuracy on subset of the Semantic-Syntactic Word Relationship test set, using word vectors from the CBOW architecture with limited vocabulary. Only questions containing words from the most frequent 30k words are used.*

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

Results

1 CPU

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
Our NNLM	20	6B	12.9	26.4	20.3
Our NNLM	50	6B	27.9	55.8	43.2
Our NNLM	100	6B	34.2	**64.5**	50.8
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

Epoch, 1 vs 3

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3
1 epoch CBOW	300	783M	13.8	49.9	33.6	0.3
1 epoch CBOW	300	1.6B	16.1	52.6	36.1	0.6
1 epoch CBOW	600	783M	15.4	53.3	36.2	0.7
1 epoch Skip-gram	300	783M	45.6	52.2	49.2	1
1 epoch Skip-gram	300	1.6B	52.2	55.1	53.8	2
1 epoch Skip-gram	600	783M	56.7	54.5	55.5	2.5

DistBelief

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

Conclusion

- Simpler is better?
- CBOW vs Continuous Skip-gram?