

LING530F: Deep Learning for Natural Language Processing (DL-NLP)

Muhammad Abdul-Mageed

muhammad.mageed@ubc.ca

Natural Language Processing Lab

The University of British Columbia

Table of Contents

1 Word2vec

- Overview of word2vec
- The Embedding Matrix
- Skip-gram model

2 Benchmarking Word Vector Models

- Syntactic Regularity
- Semantic Regularity
- Vector Offsets

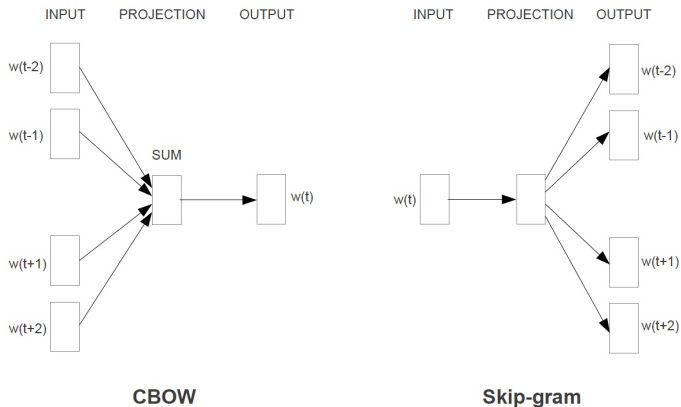


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Figure: Mikolov et al., 2013

Skip-Gram Net

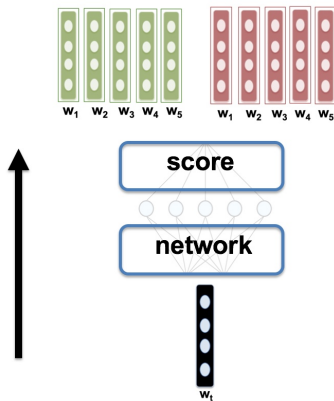


Figure: Skipgram

Vocabulary

```
1 import re
2 from nltk.corpus import stopwords
3 data=""
4 Alex bought a bike.
5 Susan rides a bike to school.
6 Alex rides a car to work.
7 Susan goes to work by car.
8 Alex goes to meetings in a suite.
9 Susan goes to parties in a suite.
10 ""
11 data=re.sub("\.", "", data)
12 words=set(data.lower().split())
13 filtered_words = [word for word in words if word not in stopwords.words('english')]
14 vocab= {k: v for v, k in enumerate(filtered_words)}
15 vocab = sorted(vocab.items(), key=lambda (k,v): v)
16 for t in vocab:
17     print(t)

('school', 0)
('alex', 1)
('susan', 2)
('car', 3)
('meetings', 4)
('work', 5)
('parties', 6)
('bike', 7)
('goes', 8)
('suite', 9)
('rides', 10)
('bought', 11)
```

One-Hot Encoding (in 4 Dimensions)

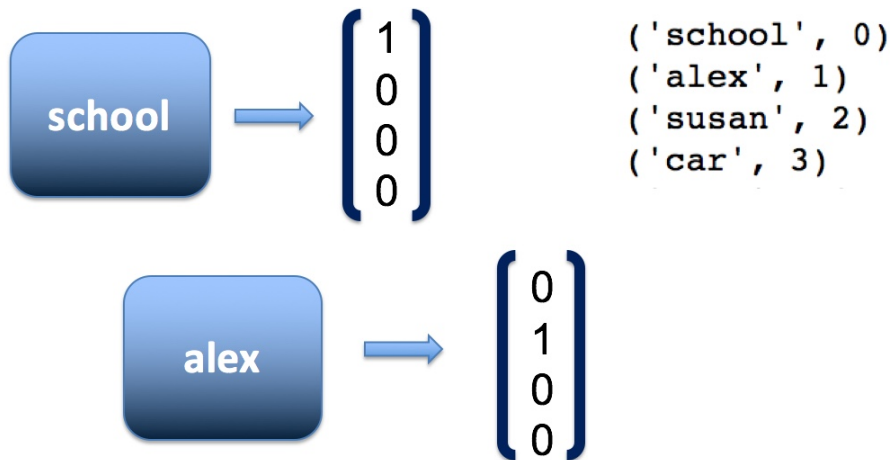


Figure: One-hot vectors in 4 dimensions

Embedding Matrix

| | alex | bike | bought | car | goes | meetings | parties | ... |
|--------|------|------|--------|-----|------|----------|---------|-----|
| school | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| alex | 1 | 1 | 0 | 1 | 1 | 1 | 0 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

Figure: An Embedding Matrix \mathbf{E} in $\mathbb{R}^{|\text{vocab}| \times |\text{context}|}$.

Embedding Matrix (Raw Counts)

| | alex | bike | bought | car | goes | meetings | parties | ... |
|---------------|-------------|-------------|---------------|------------|-------------|-----------------|----------------|------------|
| school | 32 | 57 | 755 | 354 | 908 | 4262 | 977979 | ... |
| alex | 1 | 6567 | 65674 | 154 | 7866 | 5576 | 456 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

Figure: Embedding Matrix **E** With Raw (Untuned) Counts.

Varying Context Size

susan rides a bike to school

susan rides a bike to school

susan rides a bike to school

From Raw Counts to Continuous Space

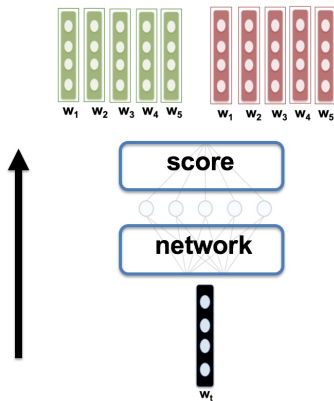


Figure: Skipgram

Skipgram Model Objective

- The **objective of the skipgram model** is to maximize the following **log-likelihood**:

1: Skipgram objective

$$\sum_{t=1}^T \sum_{c \in C_t} \log p(w_c | w_t)$$

- **Context** C_t : the set of indices of words surrounding word w_t .

Softmax as Probability of C?

- Consider that we are given a **scoring function** s which maps pairs of (word, context) to scores in \mathbb{R} .
- One possibility: Define the probability of a context word using **softmax** (as in Bengio et al., 2013):

2: Softmax scoring

$$p(w_c | w_t) = \frac{e^{w_t, w_c}}{\sum_{j=1}^W e^{s(w_t, j)}}$$

- **Not adapted to our case** as it implies we only predict one context word.
- Predicting context words can instead be framed as a **set of independent binary classification tasks**.

Learning Via Binary Classification

- Goal is to independently predict the presence (or absence) of context words.
- For the word at position t we consider all context words as **positive examples** and sample **negative examples** at random from the dictionary (**negative sampling**).

Positive and negative sample of word "butter"

- Can programmable **butter** become intelligent? (**neg sample**)
- Can programmable **machines** become intelligent? (**pos sample**)
- **Negative sampling** is brilliant as it enables us to cast the whole task as supervised learning.

Skipgram Model V: Binary Logistic Regression

- For a chosen context position c , using the binary logistic loss, we obtain the following negative log-likelihood:

3: Binary Logistic Loss I

$$\log(1 + e^{-s(w_t, w_c)}) + \sum_{n \in \mathcal{N}_{t,c}} \log(1 + e^{s(w_t, n)})$$

- where $\mathcal{N}_{t,c}$ is a set of negative examples sampled from the vocabulary.

4: Binary Logistic Loss II

- By denoting the logistic loss function $\ell : x \mapsto \log(1 + e^{-x})$, we can re-write the objective as:

$$\sum_{t=1}^T \left[\sum_{c \in C_t} \ell(s(w_t, w_c)) + \sum_{n \in N_{t,c}} \ell(-s(w_t, n)) \right]$$

Some Details of Scoring Function s

Mechanics of Scoring Function

- To **parameterize the scoring function s** between a word w_t and a context word w_c , we **use word vectors**.
- We define **for each word w in the vocabulary two vectors \mathbf{u}_w (input) and \mathbf{v}_w (output) in \mathbb{R}^d** .
- Note: In literature, you will see vectors \mathbf{u}_{wt} and \mathbf{v}_{wc} , corresponding, respectively, to w_t and w_c .
- The score can be computed as the **scalar product between word and context vectors** as $s(w_t, w_c) = \mathbf{u}_{wt}^T \mathbf{v}_{wc}$.
- Recall: The model described in this section is the **skipgram model with negative sampling**, introduced by Mikolov et al. (2013b).

But What Are Word Vectors Good For?

Linguistic Regularity

- Word representations can **capture both syntactic and semantic regularities** in language.
- Syntactic analogy questions of the form “a is to b as c is to --”.
- **Example of syntactic analogy:** “good is to better as rough is to --”

Syntactic Analogy Tests

- base/comparative/superlative forms of adjectives
- singular/plural forms of common nouns
- possessive/non-possessive forms of common nouns
- base, past and 3rd person present tense forms of verbs

Benchmarking Word Embeddings (English)

- **POS-tagged 267M words of newspaper text** with Penn. Treebank POS tags (Marcus et al., 1993)
- **Selected top 100**: comparative adjectives (JJR), plural nouns (NNS), possessive nouns (NN POS), base form verbs (VB)
- Systematically **generated analogy questions** by randomly matching each of the 100 words with 5 other words from the same category, and creating variants (see table in next slide).
- **Total test set size is 8000 example pairs**, available at [\[link\]](#).

Syntactic Regularity (For Your Reference)

| Category | Relation | Patterns Tested | # Questions | Example |
|------------|-------------------------------------|-------------------------|-------------|--------------------------|
| Adjectives | Base/Comparative | JJ/JJR, JJR/JJ | 1000 | good:better rough:____ |
| Adjectives | Base/Superlative | JJ/JJS, JJS/JJ | 1000 | good:best rough:____ |
| Adjectives | Comparative/ Superlative | JJS/JJR, JJR/JJS | 1000 | better:best rougher:____ |
| Nouns | Singular/Plural | NN/NNS, NNS/NN | 1000 | year:years law:____ |
| Nouns | Non-possessive/ Possessive | NN/NN_POS, NN_POS/NN | 1000 | city:city's bank:____ |
| Verbs | Base/Past | VB/VBD, VBD/VB | 1000 | see:saw return:____ |
| Verbs | Base/3rd Person Singular Present | VB/VBZ, VBZ/VB | 1000 | see:sees return:____ |
| Verbs | Past/3rd Person Singular Present | VBD/VBZ, VBZ/VBD | 1000 | saw:sees returned:____ |

Table 1: Test set patterns. For a given pattern and word-pair, both orderings occur in the test set. For example, if “see:saw return:____” occurs, so will “saw:see returned:____”.

Figure: Mikolov et al., 2013: Linguistic regularities in continuous space word representations.

Semantic Test Set X

- Source: SemEval-2012 Task 2, **Measuring Relation Similarity** (Jurgens et al., 2012).
- **79 fine-grained word relations** (10 for training & 69 for testing)
- Each relation is exemplified by **3 or 4 gold word pairs**
- Task: Given a group of word pairs with same relation, **order the target pairs according to the *degree* to which this relation holds.**

Example relation

- **Relation name:** OBJECT:TYPICAL ACTION
- **Relation schema:** an X will typically Y
- **Examples:** glass:break; soldier:fight
- **Analogy question:** "glass" is to "break" as "soldier" is to "?" (answer: "fight")

Measuring Relation Similarity (For Your Reference)

Measuring Relation Similarity (Jurgens et al., 2012)

- SemEval-2012 Task 2: [link]
- Goal: **identifying the degree of prototypicality for instances within a given class.**

| Subcategory | Relation name | Relation schema | Paradigms | Responses |
|-------------|-----------------------|----------------------------|--|--|
| 8(e) | AGENT:GOAL | "Y is the goal of X" | pilgrim:shrine assassin:death climber:peak | patient:health runner:finish astronaut:space |
| 5(e) | OBJECT:TYPICAL ACTION | "an X will typically Y" | glass:break soldier:fight juggernaut:crush | ice:melt lion:roar knife:stab |
| 4(h) | DEFECTIVE | "an X is is a defect in Y" | fallacy:logic astigmatism:sight limp:walk | pimple:skin ignorance:learning tumor:body |

Table 1: Examples of the three manually selected paradigms and the corresponding pairs generated by Turkers.

Figure: Jurgens et al., 2012

Sample Word Relationships

Table 1: *Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.*

| 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 |

Figure: Mikiolov et al., 2013: "Efficient Estimation...". Example semantic and syntactic relationships.

Solving Analogy Via Vector Offsets

Vector Offset Method

- To recap, both syntactic and semantic tasks are formulated as analogy questions
- **Finding:** A simple vector offset method based on cosine distance is effective in solving these questions
- **Assumption:** relationships are present as vector offsets
- Thus, in the embedding space, all pairs of words sharing a particular relation are related by the same constant offset

Vector Offset Method: How it Works

- To answer the analogy question **a:b c:d** where **d** is **unknown**, find the embedding vectors:

$$x_a, x_b, x_c,$$

and compute:

$$y = x_b - x_a + x_c$$

- **y**: the continuous space representation of the word expected to be the best answer
- **No word might exist at that exact position!**
- Search the space for the word whose embedding vector has the greatest cosine similarity to **y**:

$$w^* = \operatorname{argmax}_w \frac{x_w y}{||x_w|| ||y||}$$

- For **benchmarking**, where **d** is given from the semantic test set, simply use:

$$\cos(x_b - x_a + x_c, x_d)$$

Vector Offset Illustration

Vector Offset Example

- $\text{vector}(\text{"King"}) - \text{vector}(\text{"Man"}) + \text{vector}(\text{"Woman"}) = \text{vector}(\text{"Queen"})$.

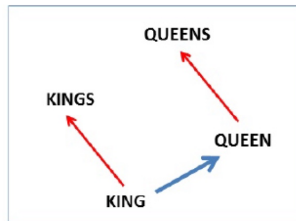
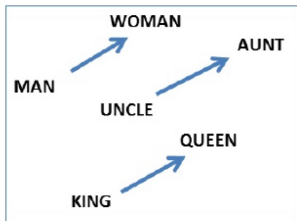


Figure: [Mikolov et al., 2013].

Sample Model Predictions

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (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 |

Figure: Mikiolov et al., 2013: "Efficient Estimation...". Example model predictions.

Other Word Embedding Models

- GloVe: [lik]
- FastText: [lik]
- **Note:** Many models created for languages other than English
- If you know a language for which no models exist, a good thing is to **create ones and share with community**