Word Embeddings

ADVISOR: PROF. AMITABHA MUKERJEE

PRANJAL SINGH
10327511
Index

1. Motivation
2. Past
3. Objectives
4. Earlier Work
   a. NLP from Scratch
   b. Bilingual Word Embeddings
   c. Vector Space Word Representation
5. Dataset
6. Results (English and Hindi)
7. Conclusion and Future Work
Motivation

- Word embeddings have the power to capture syntax and semantics both

- We have many sources of unsupervised raw data but not supervised data

- Unsupervised techniques could greatly improve existing supervised systems (Collobert et al.(2013))

Leveraging large amount of data floating around, we can improve existing systems
Past

- LSA and LDA were used to capture word embeddings (not exactly) and hence derive semantic relations.

- Most of the existing systems treat word as atomic units.

**BUT**

Words also inherit meanings which can only be defined if we represent it as a vector/combination of latent words.
Objective

To maximize probability of raw text given a context window

So for a given context window of size $c$:

$$\max \frac{1}{T} \sum_{t=1}^{T} \log p \left( w_t | w_{t+c}^t \right)$$
Earlier Work

- word2vec (Mikolov et al., 2013) learns embeddings using neural language model

- Collobert & Weston, 2011: NLP from Scratch

- Bilingual Word Representations (Zou et al. al & Manning et al., 2013)
Embeddings

Word2vec

1) CBOV

Embeddings are represented by a set of latent variables and initialized randomly. Training learns these for each word \( w_t \) in the vocabulary.

So for a given context window of size \( c \):

\[
\max \frac{1}{T} \sum_{t=1}^{T} \log p \left( w_t \mid w_{t-c}^{t+c} \right)
\]

\[
p(w_t \mid w_{t-c}^{t+c}) = \frac{\exp \left( e_w' \cdot \sum_{-c \leq j \leq c, j \neq 0} e_{w,t+j} \right)}{\sum_w \exp \left( e_w' \cdot \sum_{-c \leq j \leq c, j \neq 0} e_{w,t+j} \right)}
\]
Embeddings

Word2vec

2) Relational Constraint Model

Define $R$ as a set of relation between two words and relations have scores associated to indicate strength

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{w \in R_{w_i}} \log p(w|w_i),$$

--- They do not include scores of these relations
Interesting!!!!

A joint model:

$$\max \frac{1}{T} \sum_{t=1}^{T} \log p \left( w_t | w_{t-c}^{t+c} \right) + \frac{1}{N} \sum_{i=1}^{N} \sum_{w \in R_{w_i}} \log p \left( w | w_i \right)$$
NLP from Scratch

- Built a unified architecture for tasks such as POS tagging, Chunking, NER
- Compared against classical NLP benchmarks
- Avoided task specific engineering
- Generalize a system to handle multiple tasks
NLP from Scratch

- Learn lookup table by back propagation

- Words are mapped to $d$-dimensional vector using lookup table operation

- Lookup table returns a matrix for a given sentence
NLP from Scratch

Used entire English Wikipedia to learn word embeddings (631 million words)

Tokenized using Penn Treebank Tokenizer

The total training time was about four weeks

Window size: 11 and a Hidden layer with 100 units

**Objective**: Seek a network that computes a higher score when given a legal phrase than when given an incorrect phrase

\[
\theta \rightarrow \sum_{x \in X} \sum_{w \in D} \max \left\{ 0, 1 - f_\theta(x) + f_\theta(x^{(w)}) \right\}
\]
## NLP from Scratch

<table>
<thead>
<tr>
<th>Country</th>
<th>Word1</th>
<th>Word2</th>
<th>Word3</th>
<th>Word4</th>
<th>Word5</th>
<th>Word6</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRANCE</td>
<td>454</td>
<td>JESUS</td>
<td>XBOX</td>
<td>11724</td>
<td>SCARED</td>
<td>87025</td>
</tr>
<tr>
<td>AUSTRIA</td>
<td>GOD</td>
<td>AMIGA</td>
<td>GREENISH</td>
<td>NAILED</td>
<td>OCTETS</td>
<td></td>
</tr>
<tr>
<td>BELGIUM</td>
<td>SATI</td>
<td>PLAYSTATION</td>
<td>BLUISH</td>
<td>SMASHED</td>
<td>MB/S</td>
<td></td>
</tr>
<tr>
<td>GERMANY</td>
<td>CHRIST</td>
<td>MSX</td>
<td>PINKISH</td>
<td>PUNCHED</td>
<td>BIT/S</td>
<td></td>
</tr>
<tr>
<td>ITALY</td>
<td>SATAN</td>
<td>IPOD</td>
<td>PURPLISH</td>
<td>POPPED</td>
<td>BAUD</td>
<td></td>
</tr>
<tr>
<td>GREECE</td>
<td>KALI</td>
<td>SEGA</td>
<td>BROWNISH</td>
<td>CRIMPED</td>
<td>CARATS</td>
<td></td>
</tr>
<tr>
<td>SWEDEN</td>
<td>INDRA</td>
<td>psNUMBER</td>
<td>GREYISH</td>
<td>SCRAPED</td>
<td>KBIT/S</td>
<td></td>
</tr>
<tr>
<td>NORWAY</td>
<td>VISHNU</td>
<td>HD</td>
<td>GRAYISH</td>
<td>SCREWED</td>
<td>MEGAHertz</td>
<td></td>
</tr>
<tr>
<td>EUROPE</td>
<td>ANANDA</td>
<td>DREAMCAST</td>
<td>WHITISH</td>
<td>SECTIONED</td>
<td>MEGAPIXELS</td>
<td></td>
</tr>
<tr>
<td>HUNGARY</td>
<td>PARVATI</td>
<td>GEFORCE</td>
<td>SILVERY</td>
<td>SLASHED</td>
<td>GBit/S</td>
<td></td>
</tr>
<tr>
<td>SWITZERLAND</td>
<td>GRACE</td>
<td>CAPCOM</td>
<td>YELLOWISH</td>
<td>RIPPED</td>
<td>AMPERES</td>
<td></td>
</tr>
</tbody>
</table>
Bilingual Word Embeddings

- It proposes a method to learn bilingual embeddings rather than just monolingual embeddings.

- So it utilizes counts of MT alignments derived from Berkeley aligner to initialize monolingual embeddings of another language.

\[ W_{t-init} = \sum_{s=1}^{S} \frac{C_{ts} + 1}{C_t + S} W_s \]

- They have used the same formulation as Collobert et al. (2008) to learn embeddings except that they have used global context information as in Huang et al. (2012).
Bilingual Word Embeddings

- Their objective function captures information of both monolingual embedding and also on translation matrices, also called alignment matrices.

- They have trained on 100K-vocabulary word embeddings.

- With 500,000 iterations it took 19 days of training on an 8-core machine.

- For phrase similarity in 2 languages, they have averaged out the word embedding vectors corresponding to each word in both phrases and then taken cosine similarity to quantize amount of semantic similarity.
Dataset

- Hindi: Wikipedia text dump (279MB)
- English: Wikipedia text dump (95MB)
**Result (English)**

"boy" is to "father" as "girl" is to ...

**Top 3**

1. Mother 0.6219688653945923  
2. Grandmother 0.5560075640678406  
3. Wife 0.5442352890968323
<table>
<thead>
<tr>
<th>English Words</th>
<th>Plural Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>his</td>
</tr>
<tr>
<td>big</td>
<td>bigger</td>
</tr>
<tr>
<td>going</td>
<td>went</td>
</tr>
<tr>
<td>she</td>
<td>?</td>
</tr>
<tr>
<td>bad</td>
<td>?</td>
</tr>
<tr>
<td>being</td>
<td>?</td>
</tr>
</tbody>
</table>

- 'he' is to 'his' as 'she' is to 'her'
- 'big' is to 'bigger' as 'bad' is to 'worse'
- 'going' is to 'went' as 'being' is to 'were'
Which word doesn't go with the others?

breakfast    cereal    dinner    lunch

- cereal
रезультат (हिंदी)

<table>
<thead>
<tr>
<th>देश</th>
<th>संख्या</th>
</tr>
</thead>
<tbody>
<tr>
<td>भारत</td>
<td>0.488481163979</td>
</tr>
<tr>
<td>यूक्रेन</td>
<td>0.472263723612</td>
</tr>
<tr>
<td>मैक्सिको</td>
<td>0.461070656776</td>
</tr>
<tr>
<td>फिलीपीन्स</td>
<td>0.445656210184</td>
</tr>
<tr>
<td>कोसोवो</td>
<td>0.438328802586</td>
</tr>
<tr>
<td>कैलिफॉर्निया</td>
<td>0.437484622002</td>
</tr>
<tr>
<td>तिरुवनंतपुरम</td>
<td>0.437374174595</td>
</tr>
<tr>
<td>ओंटारियो</td>
<td>0.436686635017</td>
</tr>
<tr>
<td>सिसुआन</td>
<td>0.436174809933</td>
</tr>
<tr>
<td>लम्पुर</td>
<td>0.434365183115</td>
</tr>
</tbody>
</table>
Result (Hindi)

Odd one out

'भारत'
'रूस'
'मुम्बई'
'चीन'

'मुम्बई'
Result (Hindi)

x = similar([‘भारत’.decode(‘utf8’)], topn=5)

<table>
<thead>
<tr>
<th>प्रदेश</th>
<th>0.434905201197</th>
</tr>
</thead>
<tbody>
<tr>
<td>देश</td>
<td>0.434299349785</td>
</tr>
<tr>
<td>तिब्बत</td>
<td>0.434264868498</td>
</tr>
<tr>
<td>आन्ध्रप्रदेश</td>
<td>0.428886473179</td>
</tr>
<tr>
<td>लद्दाख</td>
<td>0.427965015173</td>
</tr>
</tbody>
</table>
Result (Hindi)

\[ x = \text{similar}([\text{‘व्यापार’}.\text{decode(‘utf8’)}], \text{topn}=5) \]

<table>
<thead>
<tr>
<th>व्यवसाय</th>
<th>0.671647787094</th>
</tr>
</thead>
<tbody>
<tr>
<td>पुनर्वापा</td>
<td>0.617935776711</td>
</tr>
<tr>
<td>वाणिज्य</td>
<td>0.612713575363</td>
</tr>
<tr>
<td>संस्थागत</td>
<td>0.61127692461</td>
</tr>
<tr>
<td>बैंकिंग</td>
<td>0.607060432434</td>
</tr>
</tbody>
</table>
Result (Hindi)

कम

0.013972 0.020021 0.005228 0.001282 -0.096880 -0.064957 -0.004378 0.057942 -0.109471 -0.052513 -0.002228
0.068519 0.117182 0.009550 0.008309 -0.035241 0.042594 0.046013 0.022055 0.033392 -0.046861 0.083555
0.003501 0.032369 -0.051409 0.042281 0.060196 0.016986 0.023544 0.014908 -0.095546 0.010151 -0.028563 -
0.079369 0.045530 -0.002945 -0.023547 -0.058014 -0.038463 0.083010 -0.028450 0.018251 0.005231 -0.006079 -
0.005987 -0.000233 0.066247 0.021251 -0.041221 -0.002379 0.064932 -0.080568 -0.113520 -0.053706 0.042745
0.021324 -0.086906 0.030630 -0.068239 -0.119651 0.027618 -0.029169 0.048726 -0.017188
Future Work

What if we **ADD** the embeddings??

Or

If we **SUBTRACT** the embeddings??

Very Big    Bigger

Such phrases and words should have greater semantic similarity

Can operations such as addition/subtraction give a better insight into such relationships (applicable for Hindi also)
Future Work

Indian Cricketer  
Sachin

Infact above phrase and word may belong to same embedding
Future Work

• The embeddings obtained could help in initializing the embeddings used in work of Collobert and Weston

• Manning et al.(2013) have used semantic information to improve word embeddings

• Collobert et al.(2008) have used large unlabeled data to do the same thing.

• Can we use syntactic or morphological information to improve word embeddings or even produce some good word embeddings?

Motivation

• Morphologically similar words have some sought of close connection between them

• e.g. morphology, phonology, etymology
References


Thank You!!!