CSE 471 MACHINE LEARNING ASSIGNMENT

Summarized Report on GloVe

Mahim Mahbub, 1505022
December 22, 2020

1 SUMMARY

1.1 INTRODUCTORY EXPLANATION

This report contains a write-up for the word embeddings of GloVe (Global Vectors for Word Representation) [1].

GloVe is an **unsupervised** learning algorithm which is used to obtain **vector representations** of words. This model captures the **global statistics** of the corpus by using a **co-occurrence matrix** of all the word pairs. Hence, the global corpus statistics are captured by this model.

The table 1.1 shows the relevant notations required to understand the working of this model.

X_{ij}	Number of times word j occurs in the context of word i
$P_{ij} = Pr(j\|i) = X_{ij}/X_i$	Probability that word j would appear in the context of word i

Table 1.1: Relevant Definitions

Now suppose i = ice and j = steam, words k related to ice but not steam eg. k = solid or cold, we can expect P_{ik}/P_{jk} to be large since P_{ik} will be large due to k being related to i.

Similarly for k related to steam eg. gas or hot, we can expect P_{ik}/P_{jk} to be small in magnitude since P_{jk} will be large due to k being related to j.

For words k related to both words i and j eg. water, the ratio P_{ik}/P_{jk} will be close to 1 as both P_{ik} and P_{jk} will end up being similar in magnitudes.

Since the ratio P_{ik}/P_{jk} depends on three words i, j, k, the most general form of a model would be $F(w_i, w_j, \bar{w_k}) = P_{ik}/P_{jk}$ where w are word vectors and $\bar{w}$ are separate context vectors.

$$F(w_i, w_j, \bar{w_k}) = \frac{P_{ik}}{P_{jk}} \tag{1.1}$$

The right hand side of the equation 1.1 i.e. P_{ik}/P_{jk} is obtained from the corpus.

After some mathematical manipulations (details in [1]), a least squares regression problem is formulated. Let V be the size of the vocabulary.

$$J = \sum_{i,j=1}^{V} (w_i^T \bar{w_j} + b_i + \bar{b_j} - \log X_{ij})^2 \tag{1.2}$$

However, this weighs all pairs with equal weights, irrespective of how frequently or rarely they occur in the corpus. Hence, a weighted function $f(X_{ij})$ is introduced in the cost function to address these issues. The complexity of the model depends on the number of non-zero entries in X due to the weighting function $f(x)$.

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \bar{w_j} + b_i + \bar{b_j} - \log X_{ij})^2 \tag{1.3}$$

The weighting functions must satisfy the following properties:

- f(0) = 0
- f(x) should be **non-decreasing** so that **rare** co-occurrences are not over-weighed.
- f(x) should be relatively **small for large values** of X, so that **frequent** co-occurrences are not over-weighed.

With these properties in mind, the authors proposed the following weighting function.

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases}$$

The authors empirically found α as $3/4$ to be a suitable value.

1.2 Training

The authors trained their model on five corpora of different sizes as shown in 1.2:

Corpus	# tokens (in billions)
2010 Wikipedia dump	1
2014 Wikipedia dump	1.6
Gigaword5	4.3
Gigaword5 + Wikipedia2014	6
Common Crawl	42

Table 1.2: Training Corpora Details

They then tokenized and lowercases each corpus with the Stanford tokenizer, to build a vocabulary of the 400,000 most frequent words, and then construct a matrix of co-occurrence counts X. While forming X, the size of the context window should be determined and it should be decided if a distinction is needed between left context and right context. In all of the cases, they used a **decreasing weighting function**, so that word pairs that are d words apart contribute $1/d$ to the total count.

For all the experiments, they had used $x_{max} = 100$, $\alpha = 3/4$, and trained the model using **AdaGrad** optimizer [2], stochastically sampling nonzero elements from X, with initial learning rate of 0.05. They ran **50** iterations for vectors smaller than **300** dimensions, and **100** iterations in other cases. They used a context of ten words to the left and ten words to the right.

The model generates two sets of word vectors, W and $\bar{W}$. And they used the sum $W + \bar{W}$ as their word vectors.

REFERENCES

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[2] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, July 2011.