# A Robustness and Effectiveness of GEM

## A.1 Robustness Test

We test the robustness of GEM by removing one non-important stop word in a sentence and computed the similarity between the original sentence and the one after removal. For example:

- original = ”The student is reading a physics book”
- removed = ”student is reading a physics book”

Stop word ”The” is removed. The cosine similarity between embeddings of the two sentences generated by GEM is 0.998. GEM assigns pretty similar embeddings for these two sentences even with the removal of stop words, especially this is a short sentence with only 7 words. More examples are:

- original = ”Someone is sitting on the blanket”
- removed = ”Someone is sitting on blanket”
- cosine similarity = 0.981

and

- original = ”A man walks along walkway to the store”
- removed = ”man walks along walkway to the store”
- cosine similarity = 0.984

These experiments prove that GEM is robust against stop words and words order.

## A.2 Effectiveness Test

We also demonstrate that GEM assign higher weights to words with more significant meanings. Consider the sentence: the stock market closes lower on Friday, weights assigned by GEM are [lower: 4.94, stock: 4.93, closes: 4.78, market: 4.62, Friday: 4.51, the: 3.75, on: 3.70]. Again, GEM emphasizes informative words like lower and closes, and diminishes stop words like the and there.

# B Proof

The novelty score ($\alpha_n$), significance score ($\alpha_s$) and corpus-wise uniqueness score ($\alpha_u$) are larger when a word $w$ has relatively rare appearance in the corpus and can bring in new and important semantic meaning to the sentence.

Following the section 3 in Arora et al. (2017), we can use the probability of a word $w$ emitted from sentence $s$ in a dynamic process to explain eq. (10) and put this as following Theorem with its proof provided below.

**Theorem 1.** *Suppose the probability that word $w_i$ is emitted from sentence $s$ is*[2]*:*

$$\mathrm{p}[w_i|\boldsymbol{c}_s] \propto (\frac{\exp(\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle)}{Z} + \exp(-(\alpha_n + \alpha_s + \alpha_u))) \tag{12}$$

*where $\boldsymbol{c}_s$ is the sentence embedding, $Z = \sum_{w_i \in \mathcal{V}} \exp(\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle)$ and $\mathcal{V}$ denotes the vocabulary. Then when $Z$ is sufficiently large, the MLE for $\boldsymbol{c}_s$ is:*

$$\boldsymbol{c}_s \propto \sum_{w_i \in s} (\alpha_n + \alpha_s + \alpha_u) \boldsymbol{v}_{w_i} \tag{13}$$

**Proof:** According to Equation (12),

$$\mathrm{p}[w_i|\boldsymbol{c}_s] = \frac{1}{N}(\frac{\exp(\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle)}{Z} + \exp(-(\alpha_n + \alpha_s + \alpha_u))) \tag{14}$$

Where $N$ and $Z$ are two partition functions defined as

$$N = 1 + \sum_{w_i \in \mathcal{V}} \exp(-(\alpha_n(w_i) + \alpha_s(w_i) + \alpha_u(w_i)))$$
$$Z = \sum_{w_i \in \mathcal{V}} \exp(\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle) \tag{15}$$

The joint probability of sentence $s$ is then

$$p[s|\boldsymbol{c}_s] = \prod_{w_i \in s} p(w_i|\boldsymbol{c}_s) \tag{16}$$

To simplify the notation, let $\alpha = \alpha_n + \alpha_s + \alpha_u$. It follows that the log likelihood $f(w_i)$ of word $w_i$ emitted from sentence $s$ is given by

$$f_{w_i}(\boldsymbol{c}_s) = \log(\frac{\exp(\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle)}{Z} + e^{-\alpha}) - \log(N) \tag{17}$$

[2] The first term is adapted from Arora et al. (2017), where words near the sentence vector $\boldsymbol{c}_s$ has higher probability to be generated. The second term is introduced so that words similar to the context in the sentence or close to common words in the corpus are also likely to occur.

$$\nabla f_{w_i}(\boldsymbol{c}_s) = \frac{\exp(\langle c_s, \boldsymbol{v}_{w_i}\rangle)\boldsymbol{v}_{w_i}}{\exp(\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle) + Ze^{-\alpha}} \tag{18}$$

By Taylor expansion, we have

$$\begin{aligned} f_{w_i}(\boldsymbol{c}_s) &\approx f_{w_i}(0) + \nabla f_{w_i}(0)^T \boldsymbol{c}_s \\ &= \text{constant} + \frac{\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle}{Ze^{-\alpha}+1} \end{aligned} \tag{19}$$

Again by Taylor expansion on $Z$,

$$\begin{aligned} \frac{1}{Ze^{-\alpha}+1} &\approx \frac{1}{1+Z} + \frac{Z}{(1+Z)^2}\alpha \\ &\approx \frac{Z}{(1+Z)^2}\alpha \\ &\approx \frac{1}{1+Z}\alpha \end{aligned} \tag{20}$$

The approximation is based on the assumption that $Z$ is sufficiently large. It follows that,

$$f_{w_i}(\boldsymbol{c}_s) \approx \text{constant} + \frac{\alpha}{1+Z}\langle \boldsymbol{c}_s, \boldsymbol{v}_{w_i}\rangle \tag{21}$$

Then the maximum log likelihood estimation of $\boldsymbol{c}_s$ is:

$$\begin{aligned} \boldsymbol{c}_s &\approx \sum_{w_i \in s} \frac{\alpha}{1+Z}\boldsymbol{v}_{w_i} \\ &\propto \sum_{w_i \in s} (\alpha_n + \alpha_s + \alpha_u)\boldsymbol{v}_{w_i} \end{aligned} \tag{22}$$

# C Experimental settings

For all experiments, sentences are tokenized using the NLTK tokenizer (Bird et al., 2009) wordpunct_tokenize, and all punctuation is skipped. $f(\sigma_j) = \sigma_j^t$ in Equation (7). In the STS benchmark dataset, our hyper-parameters are chosen by conducting parameters search on STSB dev set at $m = 7$, $h = 17$, $K = 45$, and $t = 3$. And we use the same values for all supervised tasks. The integer interval of parameters search are $m \in [5, 9]$, $h \in [8, 20]$, $L \in [35, 75]$ (at stride of 5), and $t \in [1, 5]$. In CQA dataset, $m$ and $h$ are changed to 6 and 15, the correlation term in section 2.4.2 is changed to $o_i = \|\boldsymbol{S}^T \boldsymbol{d}_i\|_2$ empirically. In supervised tasks, same as Arora et al. (2017), we do not perform principal components in supervised tasks.

# D Clarifications on Linear Algebra

## D.1 Encode a long sequence of words

We would like to give a clarification on encoding a long sequence of words, for example, a paragraph or a article. Specifically, the length $n$ of the sequence is larger than the dimension $d$ of pretrained word vectors in this case. The only part in GEM relevant to the length of the sequence $n$ is the coarse embedding in Equation (7). The SVD of the sentence matrix of the $i$th sentence is still $\boldsymbol{S} \in \mathbb{R}^{d \times n} = [\boldsymbol{v}_{w_1}, \ldots, \boldsymbol{v}_{w_n}] = \boldsymbol{U\Sigma V}^T$, where now $\boldsymbol{U} \in \mathbb{R}^{d \times d}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times n}$, and $\boldsymbol{V} \in R^{n \times n}$. Note that the $d+1$th column to $n$th column in $\boldsymbol{\Sigma}$ are all zero. And Equation (7) becomes $\boldsymbol{g}_i = \sum_{j=1}^{d} f(\sigma_j)\boldsymbol{U}_{:,j}$. The rest of the algorithm works as usual. Also, Gram-Schmidt (GS) process is computed in the context window of word $w_i$, and the length of context window is set to be $2m+1 = 17$ in STS benchmark dataset and supervise downstream tasks. That is, GS is computed on 17 vectors, and 17 is smaller than the dimension $d$. Therefore, GS is always validate in our model, independent with the length of the sentence.

## D.2 Sensitivity to Word Order

Although utilizing Gram-Schmidt process (GS), GEM is insensitive to the order of words in the sentence, explained as follows. The new semantic meaning vector $q_i$ computed from doing GS on the context window matrix $\boldsymbol{S}^i$ is independent with the relative order of first $2m$ vectors. This is because in GEM $w_i$ (the word we are calculating weights for) is always shifted to be the last column of $\boldsymbol{S}^i$. And weighting scheme in GEM only depends on $\boldsymbol{q}_i$. Therefore, weight scores stay the same for $w_i$.