Project Report: Word Embeddings

Yansen Han

May 8, 2019

Contents

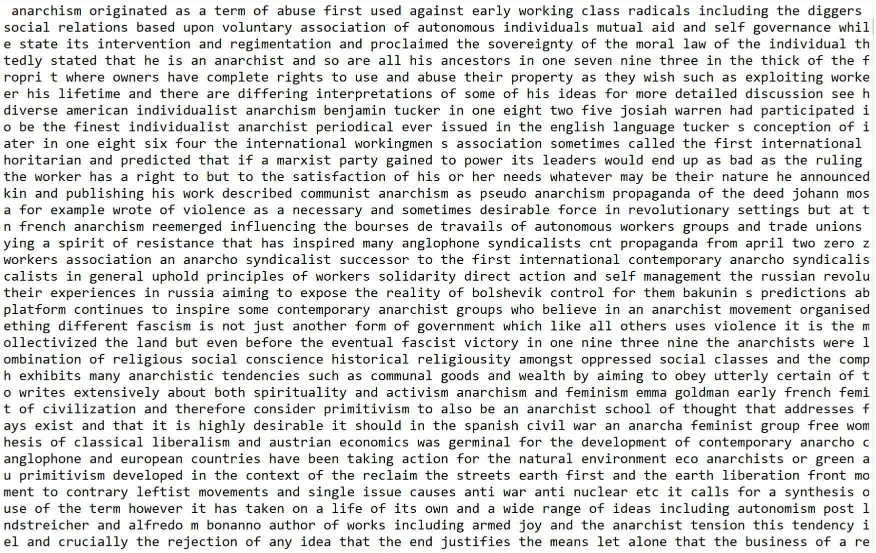
Part I

Practical Experiments in Word Embeddings (Skip-Gram and GloVe)

1 Dataset and Project Architecture

1.1 Descriptions of Dataset

My training dataset is the classical dataset, text8, which is usually used to train word embeddings model. The following figure is a view of text8.zip [1], which is approximately 30 MB.



```
 anarchism originated as a term of abuse first used against early working class radicals including the diggers
social relations based upon voluntary association of autonomous individuals mutual aid and self governance whil
e state its intervention and regimentation and proclaimed the sovereignty of the moral law of the individual th
tedly stated that he is an anarchist and so are all his ancestors in one seven nine three in the thick of the f
ropri t where owners have complete rights to use and abuse their property as they wish such as exploiting worke
er his lifetime and there are differing interpretations of some of his ideas for more detailed discussion see h
diverse american individualist anarchism benjamin tucker in one eight two five josiah warren had participated i
o be the finest individualist anarchist periodical ever issued in the english language tucker s conception of i
ater in one eight six four the international workingmen s association sometimes called the first international
horitarian and predicted that if a marxist party gained to power its leaders would end up as bad as the ruling
the worker has a right to but to the satisfaction of his or her needs whatever may be their nature he announced
kin and publishing his work described communist anarchism as pseudo anarchism propaganda of the deed johann mos
a for example wrote of violence as a necessary and sometimes desirable force in revolutionary settings but at t
n french anarchism reemerged influencing the bourses de travails of autonomous workers groups and trade unions
ying a spirit of resistance that has inspired many anglophone syndicalists cnt propaganda from april two zero z
workers association an anarcho syndicalist successor to the first international contemporary anarcho syndicalis
calists in general uphold principles of workers solidarity direct action and self management the russian revolu
their experiences in russia aiming to expose the reality of bolshevik control for them bakunin s predictions ab
platform continues to inspire some contemporary anarchist groups who believe in an anarchist movement organised
ething different fascism is not just another form of government which like all others uses violence it is the m
ollectivized the land but even before the eventual fascist victory in one nine three nine the anarchists were l
ombination of religious social conscience historical religiousity amongst oppressed social classes and the comp
h exhibits many anarchistic tendencies such as communal goods and wealth by aiming to obey utterly certain of t
o writes extensively about both spirituality and activism anarchism and feminism emma goldman early french femi
t of civilization and therefore consider primitivism to also be an anarchist school of thought that addresses f
ays exist and that it is highly desirable it should in the spanish civil war an anarcha feminist group free wom
hesis of classical liberalism and austrian economics was germinal for the development of contemporary anarcho c
anglophone and european countries have been taking action for the natural environment eco anarchists or green a
u primitivism developed in the context of the reclaim the streets earth first and the earth liberation front mo
ment to contrary leftist movements and single issue causes anti war anti nuclear etc it calls for a synthesis o
use of the term however it has taken on a life of its own and a wide range of ideas including autonomism post l
ndstreicher and alfredo m bonanno author of works including armed joy and the anarchist tension this tendency i
el and crucially the rejection of any idea that the end justifies the means let alone that the business of a re
```

Figure 1: A Glance of text8.zip File

1.2 Project Architecture

My experiments are mainly about the implements of word embedding models (Skip-gram model and GloVe model). The architecture to achieve this goal is shown in figure 2.

2 Descriptions of Skip-gram and GloVe Model

Both skip-gram model and GloVe model are used to construct word embeddings. In the paper of GloVe model (Pennington et al. (2014)), the auther point out that skip gram model can do very well in word analogy task, but it poorly utilize the statistics of corpus since they train on separate local context windows instead of global co-occurrence counts.

[1] http://mattmahoney.NET/dc/text8.zip

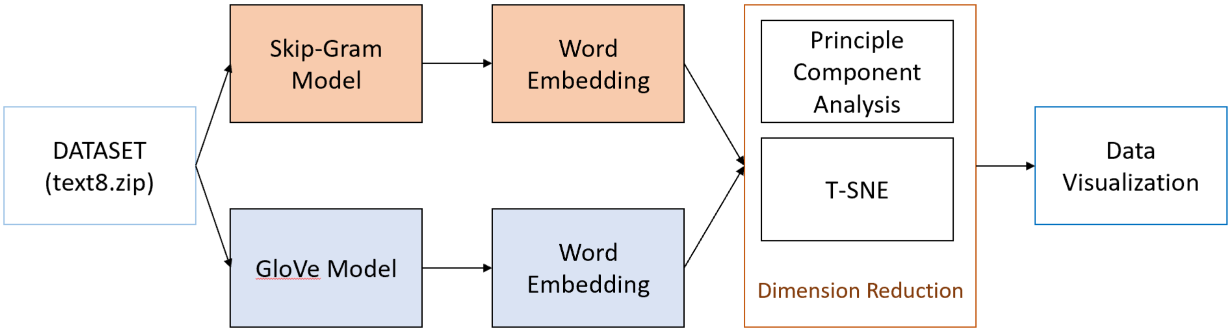


Figure 2: Project Architecture

2.1 Skip-gram model and GloVe model

The Skip-gram model: takes advantage of one central word to forecast the words surrounding the central word. This idea is constructed in a formula:

- Approximation function:

$$P(w_j|w_i) = Q_{ij} = \frac{\exp{(w_i^T \tilde{w}_j)}}{\sum_{k=1}^{V} \exp{(w_i^T \tilde{w}_k)}}$$

- Loss function:

$$J = -\sum_{i \in corpus, j \in context(i)} \log Q_{ij}$$

The Glove Model: X_{ij} tabulate the number of time word j occurs in the context of word i. Let $X_i = \sum_k X_{ik}$ be the number of times any word appears in the context of word i. Finally, let $P_{ij} = P(j|i) = X_{ij}/X_i$ be the probability that word j appear in the context of word i.

- Approximation function:

$$log(X_{ik}) = w_i^T \tilde{w}_k + b_i + \tilde{b}_k$$

- Loss fucntion:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

2.2 The Relationship between Skip-gram model and GloVe model

We derive the relationship from the loss function of Skip-gram model.

$$J = -\sum_{i \in corpus, j \in context(i)} \log Q_{ij}$$

Researchers thought this loss function is not very efficient. Therefore, they modify the loss function and make some simplification.

$$\begin{aligned} J &= -\sum_{i=1}^{V}\sum_{j=1}^{V} X_{ij}\log Q_{ij} \\ &= -\sum_{i=1}^{V} X_i \sum_{j=1}^{V} P_{ij}\log Q_{ij} \\ &= \sum_{i=1}^{V} X_i H(P_i, Q_i) \end{aligned}$$

After obtaining the entropy form of loss function, the researchers further modify it.

$$\begin{aligned} \hat{J} &= \sum_{i,j} X_i(\log \hat{P}_{ij} - \log \hat{Q}_{ij})^2 \\ &= \sum_{i,j} X_i(w_i^T \tilde{w}_j - \log X_{ij})^2 \\ &\xrightarrow{modify} \sum_{i,j} f(X_i)(w_i^T \tilde{w}_j - \log X_{ij})^2 \end{aligned}$$

Finally, we unexpectedly get a formula similar to the loss function of GloVe model.

3 Experiments Results

3.1 Results Visualization

In the first glance of my result, I find that the words close to each other don't have similar meaning. Therefore, in order to give a good reference for comparison. I would like to show the GloVe model[2] from goolge (see Figure 3).

As you can see from the Figure 3, the words close to each other also don't have the same meaning, therefore, I need to find another way to check my model and results. At first, I will compare the results from google and mine in Figure 4.

In the Figure 4, we can clearly see that the relative position of 'western', 'east' and 'west' are similar in both of the figures. I, therefore, think my result is not that bad. In the following, I will further illustrate my results using the way in the paper of GloVe model. The word embeddings obtained from skip-gram and glove model are firstly processed by PCA and T-SNE, two dimension reduction techniques. After reducing their dimensions to a plane vector, they can be shown in a plane (See Figure 5 and 6).

3.2 Results Analysis

From these two visualization figures, we can clearly find that the words with similar meaning are not close to each other in two figures. After seeing the result from google, I think this

[2] https://colab.research.google.com/github/mdda/deep-learning-workshop/blob/master/notebooks/5-RNN/3-Text-Corpus-and-Embeddings.ipynbscrollTo=LusgTEmtqTK5

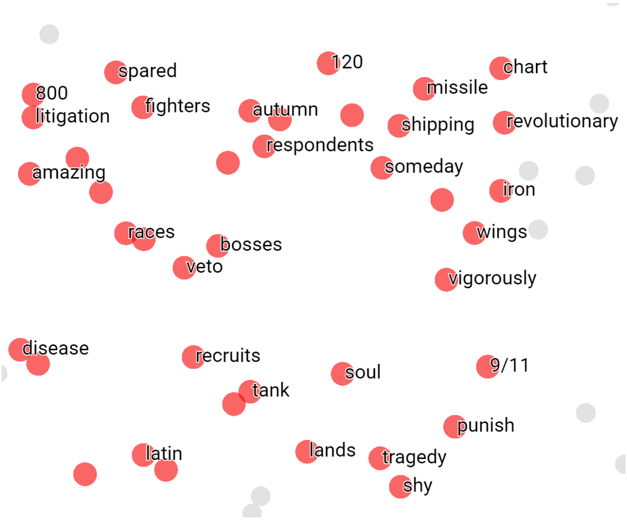
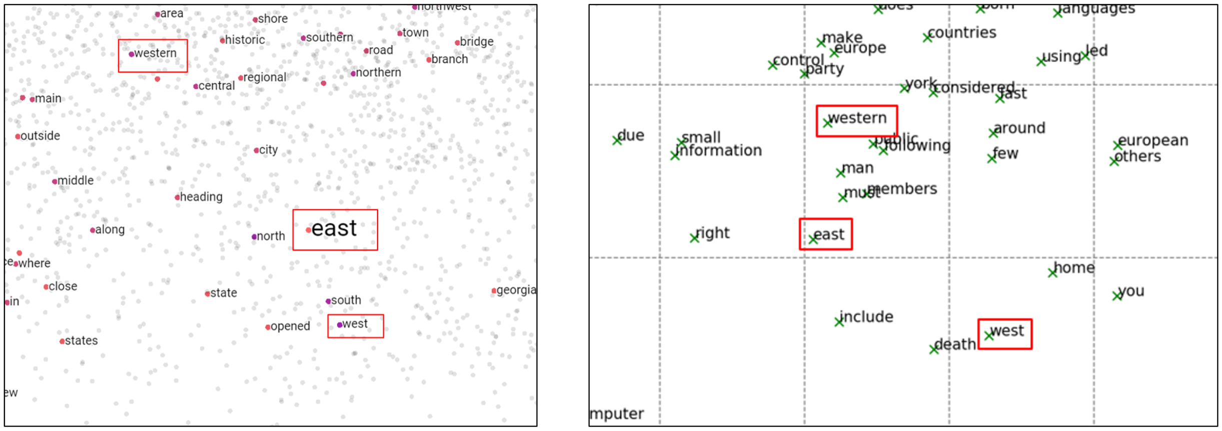


Figure 3: A Part of Visualization Result of Google's Glove Implementation



Google's Result

My Result

Figure 4: Comparison of Google's Result with Mine

may result from the dimension reduction, because this process will cause a huge amount of information loss. To sovle this problem, We can apply more efficient dimension reduction techniques. As far as i know, TSNE is the most widely used manifold reduction method, so it is really hard to say the visualization would be better after applying other dimension reduction methods.

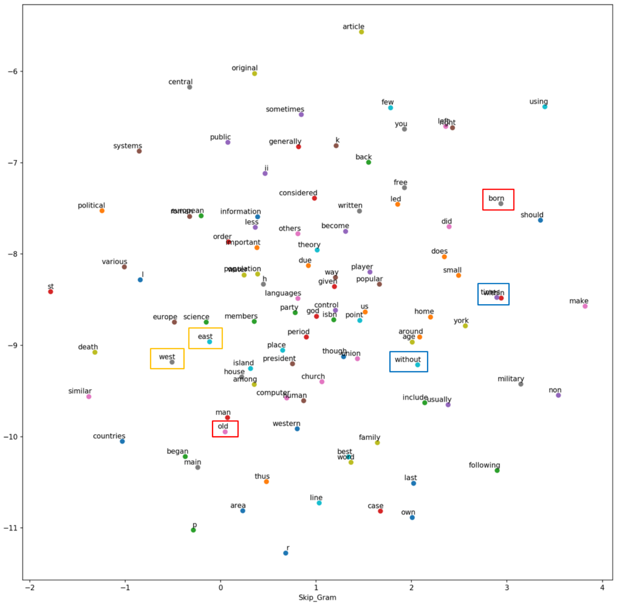


Figure 5: Visualization of Word Embedding from Skip-Gram Model

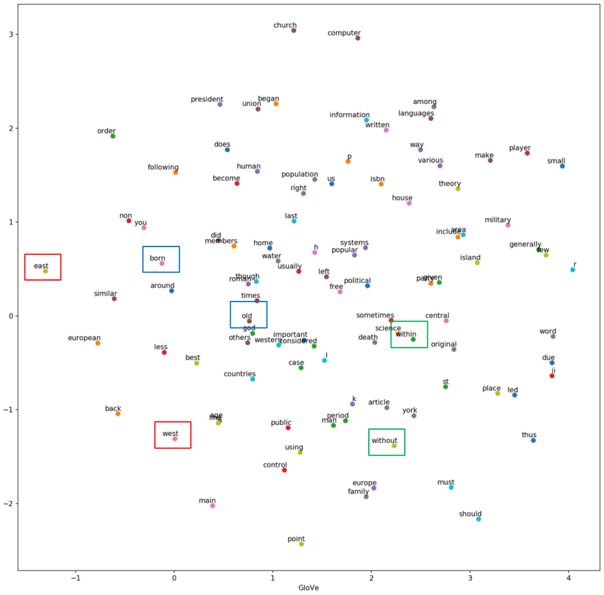


Figure 6: Visualization of Word Embedding from GloVe Model

Part II

Experiments on ChatBot

Recently, FinTech becomes unexpectedly popular and an importance usage in FinTech is ChatBot, because people hope to design a robo-advisor to chat with clients and gain some information about them. In order to practice for my future career, I will refer to some online codes and tutorials and practice my skills in construct chatbot.

4 Dataset and Data Preprocessing

4.1 Descriptions of Dataset

The training dataset is STC3 [3], which is about 600 MB.

The dataset is constructed from Weibo posts and replies/comments. More than 1 million Weibo post-response pairs will be provided to participants for training their models.

The test dataset consists of about 5000 posts while 100-200 of the posts will be manually assessed, and for each post, at most 3 emotion classes will be manually specified to indicate the emotion class of a generated response.

The dataset will include labels of each post and response. These labels are for reference only, and they are obtained by a simple classifier that is based on a bidirectional LSTM model. The classifier was trained on the data from the NLPCC Emotion Classification Challenge . The accuracy of our classifier for six-way classification is about 64%. In other words, the emotion label of these data is NOISY.

The correspondence between the label and the emotion class can be seen as follows:

0: Other 1: Like 2: Sadness 3: Disgust 4: Anger 5: Happiness

The training dataset looks like:

$[[[post, post\_label], [response, response\_label]], [[post, post\_label], [response, response\_label]], ...]$.

4.2 Data Preprocessing

In order to construct a chatbot to comfort people, I extract those sentence pairs with the emotions of post sentence are in class 1, 5. Besides, I also include the emotion class 0 for enlarging the training dataset.

After processing, the data are in the form shown in the figure.

5 Basics of Chatbot

Generally, Chatbot can be divided into three types according to their types of model:

- Rule-based model
- Retrieval-based model

[3] $http://coai.cs.tsinghua.edu.cn/hml/challenge/dataset\_description$

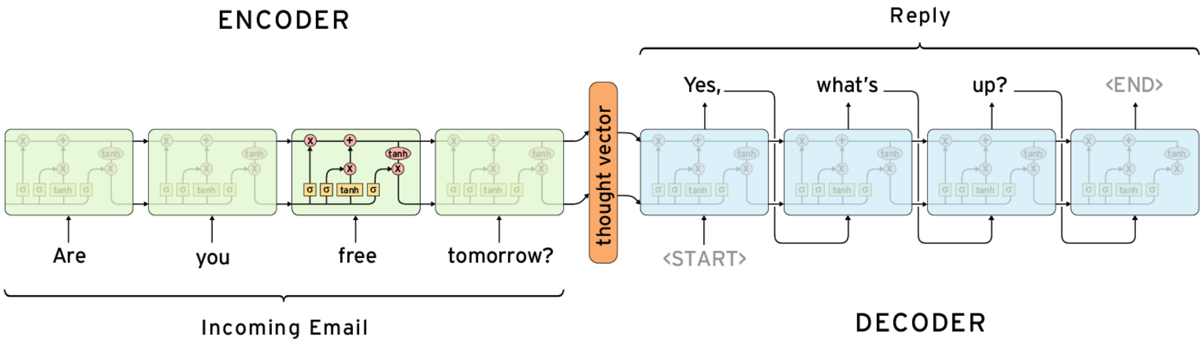


Figure 7: Sequence to Sequence model for Chatbot

- Generative model

Different kinds of model will result in remarkably different performance. In the following, I will illustrate those three kinds of models.

5.1 Chatbot on Rule-based Model

This kind of chatbot answers people's questions based on some pre-designed rules. Those rules tell chatbot what they should say when meeting some specific words. For example, if ”weather” in user's query, chatbot would say ”The weather is very good today.” This kind of chatbot can be finished very quickly, but it performs not very well in some situations.

5.2 Chatbot on Retrieval-based Model

In effect, retrieval-based model is an QA system, it will compare the inputs with the questions in the QA system, and then output the answers of most similar question in QA system.

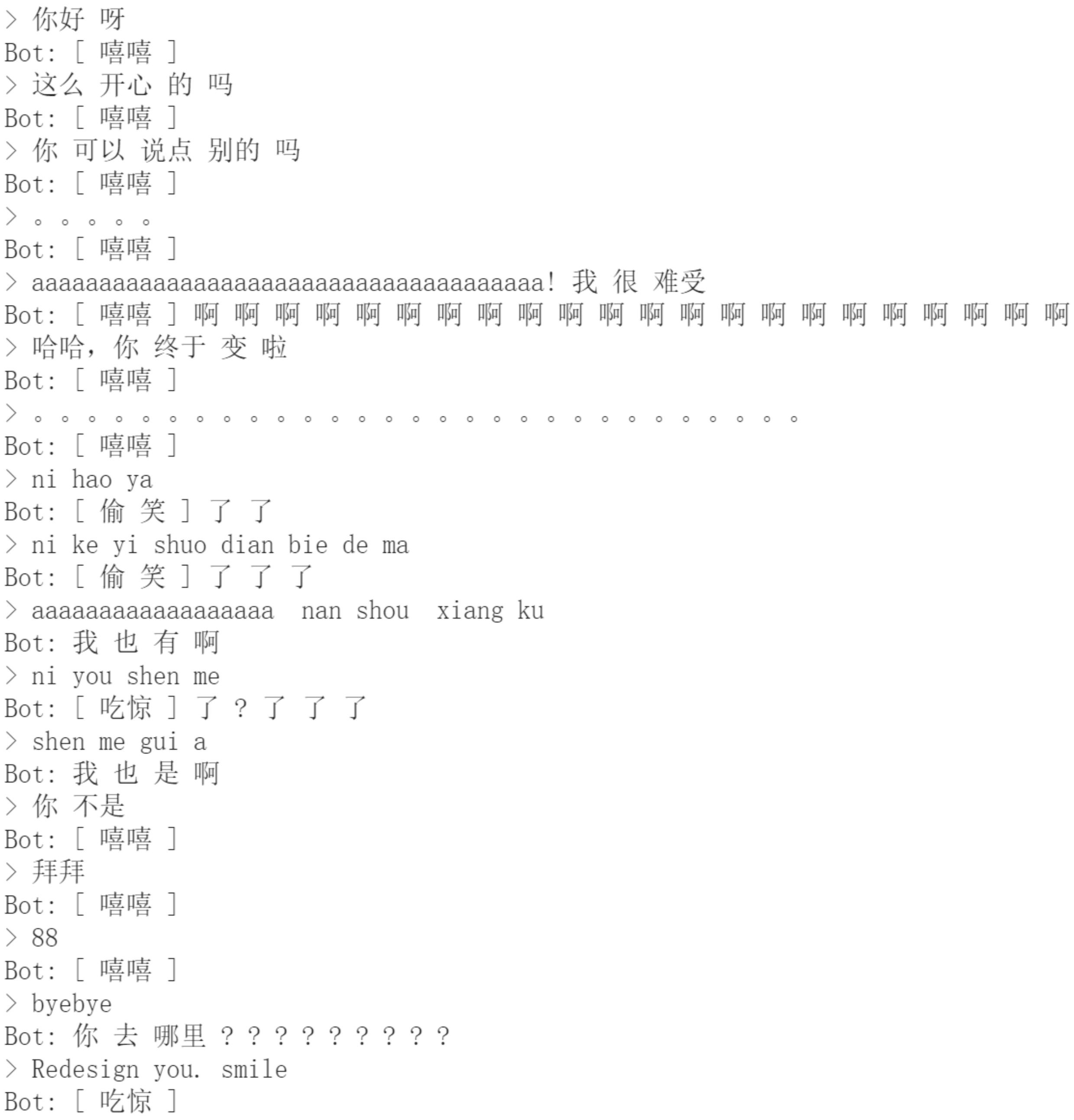
5.3 Chatbot on Generative Model

Generative model is more exciting, even though it is hard to implement this kind of chatbot. Currently, practitioners utilize sequence to sequence with attention model to implement chatbot. This kind of chatbot is not designed for specific field. It instead is to achieve some daily conversations. In order to deeply understand this kind of model, I put on a figure (Figure 7) to show the architecture of this kind of model.

6 Experiments Results

I have trained the model for 10000 iterations, but the loss of my model is still around 4.

After finishing the training , I tested the trained chatbot. The figure 8 is the conversation with my stupid chatbot.



```
> 你好 呀
Bot: [ 嘻嘻 ]
> 这么 开心 的 吗
Bot: [ 嘻嘻 ]
> 你 可以 说点 别的 吗
Bot: [ 嘻嘻 ]
> 。 。 。 。 。
Bot: [ 嘻嘻 ]
> aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa! 我 很 难受
Bot: [ 嘻嘻 ] 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊 啊
> 哈哈，你 终于 变 啦
Bot: [ 嘻嘻 ]
> 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。
Bot: [ 嘻嘻 ]
> ni hao ya
Bot: [ 偷 笑 ] 了 了
> ni ke yi shuo dian bie de ma
Bot: [ 偷 笑 ] 了 了 了
> aaaaaaaaaaaaaaaaaa  nan shou  xiang ku
Bot: 我 也 有 啊
> ni you shen me
Bot: [ 吃惊 ] 了 ？ 了 了 了
> shen me gui a
Bot: 我 也 是 啊
> 你 不是
Bot: [ 嘻嘻 ]
> 拜拜
Bot: [ 嘻嘻 ]
> 88
Bot: [ 嘻嘻 ]
> byebye
Bot: 你 去 哪里 ？ ？ ？ ？ ？ ？ ？ ？ ？
> Redesign you. smile
Bot: [ 吃惊 ]
```

Figure 8: Conversations with My Stupid Chatbot

Part III
Research Review on Word Embedding and Related Fields

7 Researches on Word Embeddings

This kind of researches aims to encode the words in a representative form. Previously, researchers have tried order form and one-hot form. However, these two representative forms are not very efficient, because they can not reflect some information of the word itself. In order to taking a more informative form to represent the words, researchers starts to utilize some methods similar to autoencoder. In this section, several famous approaches will be illustrated.

7.1 A Probabilistic Model for Learning Multi-prototype Word Embeddings (Tian et al. (2014))

This paper is to address the polysemy of words. Before this paper, researchers firstly train single prototype word representations through a multi-layer neural network with the assumption that one word only yields single word embedding; then, they identify multiple word embedding for each polysemous word by clustering all its context window features, which usually computed as the average of single prototype embeddings of its neighboring words in the context window. The methods above are usually sensitive to the clustering algorithms and has scalable problems.

In this paper, the author employed a latent variable to denote the classes which the word belongs to. The formula is:

$$\begin{aligned} p(w_O|w_I) &= \sum_{i=1}^{N_{wI}} P(w_O|h_{w_I}=i,w_I)P(h_{w_I}=i|w_I) \\ &= \sum_{i=1}^{N_{w_I}} \frac{exp(U_{w_O}^T V_{w_I,i})}{\sum_{w\in W} exp(U_w^T V_{w_I,i})} P(h_{w_I}=i|w_I) \end{aligned}$$

Finally, they use Expectation-Maximization method to estimation their proposed model.

7.2 Glove: Global vectors for word representation(Pennington et al. (2014))

In this paper, the researchers point out that Methods like skip-gram may do well on the analogy task, but they poorly utilize the statistics of the corpus since they train on separate local context windows instead of on global co-occurrence counts.

The Glove Model: X_{ij} tabulate the number of time word j occurs in the context of word i. Let $X_i = \sum_k X_{ik}$ be the number of times any word appears in the context of word

i. Finally, let $P_{ij} = P(j|i) = X_{ij}/X_i$ be the probability that word j appear in the context of word i.

- Objective function:
$$log(X_{ik}) = w_i^T \tilde{w}_k + b_i + \tilde{b}_k$$

- Loss fucntion:
$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

The Relationship to Skip-gram Model:

- **Skip-gram model:**
$$P(w_j|w_i) = Q_{ij} = \frac{\exp{(w_i^T \tilde{w}_j)}}{\sum_{k=1}^{V} \exp{(w_i^T \tilde{w}_k)}}$$

- **Loss function of Skip-gram model:**
$$J = - \sum_{i \in corpus, j \in context(i)} \log Q_{ij}$$

  However, the researchers thought this loss function is not very efficient. Therefore, they make modify the loss function and make some simplification.

$$\begin{aligned} J &= - \sum_{i=1}^{V} \sum_{j=1}^{V} X_{ij} \log Q_{ij} \\ &= - \sum_{i=1}^{V} X_i \sum_{j=1}^{V} P_{ij} \log Q_{ij} \\ &= \sum_{i=1}^{V} X_i H(P_i, Q_i) \end{aligned}$$

  After obtaining the entropy form of loss function, the researchers further modify it.

$$\begin{aligned} \hat{J} &= \sum_{i,j} X_i (\log \hat{P}_{ij} - \log \hat{Q}_{ij})^2 \\ &= \sum_{i,j} X_i (w_i^T \tilde{w}_j - \log X_{ij})^2 \\ &\xrightarrow{modify} \sum_{i,j} f(X_i)(w_i^T \tilde{w}_j - \log X_{ij})^2 \end{aligned}$$

The final formula looks very similar to the loss function of Glove except for two constants.

Ideas

- Combine the EM skip-gram model for polysemy and glove model

7.3 A Unified Model for Word Sense Representation and Disambiguation (Chen et al. (2014))

This paper proposed a unified model for joint word sense representation and disambiguation. The basic idea is that both word sense representation and word sense disambiguation will benefit from each other when training models. Their model contains three-stage process:

Stage 1: Initializing word vectors and sense vectors

- **Initializing word vectors:** Using Skip-gram model.

$$\frac{1}{T}\sum_{t=1}^{T}(\sum_{-k\leq j\leq k, j\neq 0}\log p(w_{t+j}|w_t))$$

$$p(w_{t+j}|w_t) = \frac{\exp(vec'(w_{t+j})^T vec(w_t))}{\sum_w \exp(vec'(w)^T vec(w_t))}$$

  where k is the size of the training window, T is the cardinality of the vocabulary. In this step, we will obtain the representation of word $vec(w)$. Before illustrating next step, I will introduce the notations. The unlabeled texts are denotes as R, and the vocabulary of the tests is denoted as W. For a word w in W, w_{s_i} is the ith sense in WordNet WN. Each sense w_{s_i} is the ith sense in WordNet WN. Each sense w_{s_i} has a gloss(w_{s_i}) in WN. The word embedding of w is denoted as $vec(w)$, and the sense embedding of its ith sense w_{s_i} is denoted as $vec(w_{s_i})$.

- **Initializing sense vectors:** The basic idea of the sense vector initialization is to represent the sense by using the similar words in the gloss. From the content words in the gloss, we select those words whose cosine similarities with the original word are larger than a similarity threshold θ. Formally, for each sense w_{s_i} in WN, We first define a candidate set from $gloss(w_{s_i})$

$$cand(w_{s_i}) = \{u|u \in gloss(w_{s_i}), u \neq w, POS(u) \in CW, cos(vec(w), vec(u)) > \sigma\}$$

  The average of the word vectors in $cand(w_{s_i})$ is used as the initialization value of the sense vector $vec(w_{s_i})$:

$$vec(w_{s_i}) = \frac{1}{|cand(w_{s_i})|}\sum_{u\in cand(w_{s_i})} vec(u)$$

Stage 2: Performing Word Sense Disambiguation

The authors design two algorithms, L2R algorithm and S2C algorithm, for word sense disambiguation based on the sense vectors.

- **Context vector initialization:** Similar to the initialization of sense vectors, we use the average of all of the content words' vectors in a sentence as the initialization vector of context.

$$vec(context) = \frac{1}{|cand(S)|}\sum_{u\in cand(S)} vec(u)$$

  where $cand(S)$ is the set of content words $cand(S) = \{u|u \in S, POS(u) \in CW\}$.

- **Ranking words** For S2C, we rank the words based on the ascending order of $|Sense_{WN}(w_i)|$.
- **Word sense disambiguation:** For each word in L, we can compute the cosine similarities between the context vector and its sense vectors. We choose the sense that yields the maximum cosine similarity as its disambiguation result.

Stage 3: Learning Sense Vectors from Relevant Occurrences

Our training goal is to train the vector representations that are not only good at predicting its context words but are also good at predicting its context words' senses.

Given the disambiguation result $M(w_1), ..., M(w_T)$, the training goal is modified to:

$$\frac{1}{T}\sum_{i=1}^{T}(\sum_{j=-k}^{k}\log\{p(w_{t+j}|w_t)p(M(w_{t+j}|w_t))\})$$

where k is the size of the training window.

7.4 CBOW Is Not All You Need: Combining CBOW With The Compositional Matrix Space Model(Mai et al. (2019))

This paper is aimed to improve the performance of CBOW by solving the problem: CBOW doesn't consider the order of words. In this paper, the researchers transform the word represented in vectors into matrix, because the multiplicative operation of matrix doesn't satisfy the communicative law and they call their model CMOW (continual multiplication of words). By comparing CBOW and CMOW, the researchers said that CBOW is good at memorizing the sentence content, while CMOW is more powerful in linguistics order. Therefore. they concatenate both of the word embeddings and train them at the same time. In addition to the hybrid model, they also proposed a different objective function from the one of CBOW and compare their proposed initialization methods and Glorot initialization methods on their hybrid model. In the following, I will give a brief introduction of their work.

- Training Objective:

$$\log\sigma(v_{w_O}^T enc_\Delta^E(s)) + \sum_{i=1}^{k} E_{w_i\sim P_n(w)}[\log\sigma(-v_{w_i}^T enc_\Delta^E(s))]$$

- Initialization: In order to avoid vanishing values, they proposed the initial embedding matrix as a random deviation from identity matrix:

$$E[w] = \begin{bmatrix} N(0,0.1) & \dots & N(0,0.1) \\ \vdots & \ddots & \vdots \\ N(0,0.1) & \dots & N(0,0.1) \end{bmatrix} + \mathbf{I}_d$$

- Hybrid CBOW-CMOW model:

$$\log\sigma(v_{w_O}^T[enc_{\Sigma}^{E_1}(s); enc_{\Pi}^{E_2}(s)]) + \sum_{i=1}^{k} E_{w_i\sim P_n(w)}[\log\sigma(-v_{w_i}^T[enc_{\Sigma}^{E_1}(s); enc_{\Pi}^{E_2}(s)])]$$

7.5 Embedding Text in Hyperbolic Spaces (Dhingra et al. (2018))

In this paper, the researchers utilize non-Euclid geometry to improve the traditional word embeddings model. In the following, I will briefly introduction the idea of this paper.

The final goal is to learn a function $f : S \rightarrow B^d$ that maps objects from a sentences set to the Poincare ball B^d, which is a hyperbolic space. The distance defined in this space is $d(u, v) = cosh^{-1}(1 + 2\frac{||u-v||^2}{(1-||u||^2)(1-||v||^2)})$. Traditionally, practitioners only some deep neural network, like GRU, LSTM, to produce representations in arbitrary subspaces of $\mathcal{R}^{d'}$. This paper introduce a reparameterization that maps $\mathcal{R}^{d'}$ to $\mathcal{B}^d$ which can also be used on top of any existing encoder. The reparameterization involves computing a direction vector v and a norm magnitude p from Euclid word embeddings $e(s)$.

After introducing the geometrical basics, the researchers proposed their improvements in three embedding models.

- **Non-parametric supervised embeddings**: The dataset is represented by a set of tuples $\mathcal{D} = \{(u, v)\}$, where each pair (u, v) denotes that u is a parent of v. Since u and v come from a fixed vocabulary of objects, they use a look up table L as the encoder, i.e. $e(u) = L(u)$. They use the loss function, which uses negative samples $N(u) = \{v : (u, v) \notin \mathcal{D}, v \neq u\}$ to maximize distance between embeddings of unrelated objects:

$$\mathcal{L} = -\sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-d(u,v)}}{\sum_{v'\in N(u)\{v\}} e^{-d(u,v')}}$$

- **Non-parametric unsupervised word embeddings**: Because CBOW models, like word2vec and glove, compute the cooccurrence of pairs of words within a fixed window size. Therefore, the researchers construct a cooccurrence graph $\mathcal{G} = \{(w, v)\}$. They preserve the frequency information by allowing repeated edges in $\mathcal{G}$: each pair (w, v) occurs f^c times in $\mathcal{G}$. After constructing the cooccurrence graph, they train the transition function as the point above.

- **Parametric unsupervised sentence embeddings**:

$$c_t = \frac{1}{2K}\sum_{k=1}^{K} v'_{w_{t-k}} + v'_{w_{t+k}}$$

$$P(w_t|w_{\neq t, f_\theta(s_i)}) \propto \exp\{-\lambda_1 d(v_{w_t}, f_\theta(s_i)) - \lambda_2 d(v_{w_t}, c_t)\}$$

To ensure that $v_{w_t}, f_\theta(s_i), c_t \in \mathcal{B}^d$, they use the following parameterization:

$$\phi_{dir}(x) = W_1^T x, \phi_{norm}(x) = W_2^T x$$

where $x = \{\hat{v_{w_t}}, \hat{c}_t, \hat{f}_\theta(s_i)\}$, $\hat{v_{w_t}}$ is the Euclid output embedding of word w_t, from a look up table; $\hat{c}_t$ is the Euclidean local context embedding; and $\hat{f}_\theta(s_i)$ is a bidirectional GRU encoder over the word of s_i:

$$h_T^f = \overrightarrow{GRU}(s_i), h_T^b = \overleftarrow{GRU}(s_i)$$

$$\hat{f}_\theta(s_i) = h_T^f || h_T^b$$

The loss function is $-\sum_t \log P(w_t|w_{\neq t, f_\theta(s_i)})$

7.6 Ideas on Improving CBOW

- Implement CMOW in Hyperbolic space, which intuitively will consider the heirarchy structure and the words order.

$$c_t = enc_{\Pi}^{E}(s)$$
$$P(w_t|w_{\neq t, f_\theta(s_i)}) \propto \exp\{-d(v_{w_t}, c_t)\}$$

To ensure that $v_{w_t}, f_\theta(s_i), c_t \in \mathcal{B}^d$, they use the following parameterization:

$$\phi_{dir}(x) = W_1^T x, \phi_{norm}(x) = W_2^T x$$

where $x = \{\hat{v_{w_t}}, \hat{c_t}\}$, $\hat{v_{w_t}}$ is the Euclid output embedding of word w_t, from a look up table; $\hat{c_t}$ is the Euclidean local context embedding;

$$\log \sigma(d(v_{w_t}, [enc_{\Sigma}^{E_1}(s); enc_{\Pi}^{E_2}(s)])) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)}[\log \sigma(-d(v_{w_i}, [enc_{\Sigma}^{E_1}(s); enc_{\Pi}^{E_2}(s)]))]$$

8 Researches on Contextual Word Embeddings

Recently, BERT hit the board in all of NLP tasks and stir up the interests of all of the NLP researchers. BERT is actually one of the contextual word embeddings. In order to get deeper with contextual word embeddings, we will read and introduce some related research works in this section.

8.1 BERT: Pre-training of Deep Bidirectional Transformers for Language UnderstandingDevlin et al. (2018)

BERT addresses the unidirectional constraints in ELMO and OpenAI GPT by proposing a new pre-training objective: the ”masked language model”. The masked language model randomly masks some of the tokens from input, and the objective is to predict the original vocabulary id of the masked word based only on its contents.

In the following picture, we can clearly see that BERT is bidirectional, while GPT and ELMo are unidirectional structure. The embedding in BERT has three parts, token embed-

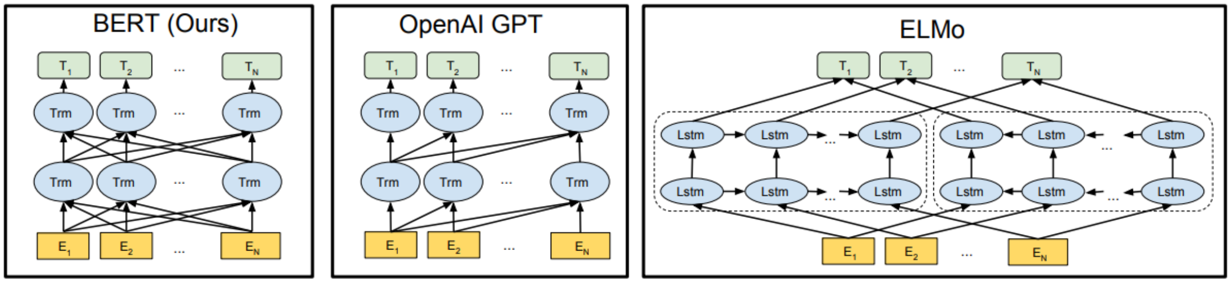


Figure 9: Different Pre-Training Models

dings, segment embeddings and position embedding. Token embedding is used to embed the words. Segment embeddings is used to denote the position of each sentence, while position embedding is to embed the position of each word.

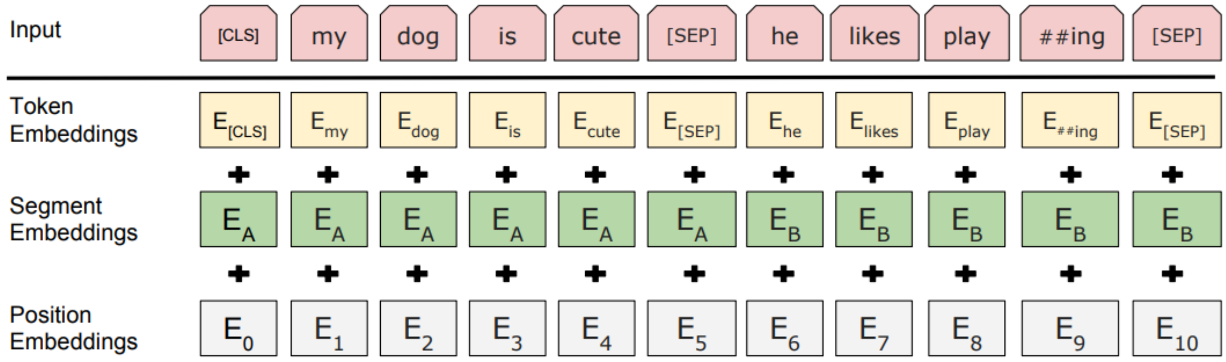


Figure 10: BERT Input Representation

As to the pre-training tasks, the researchers trained BERT on two tasks, Masked LM and next sentence prediction.

8.2 Starspace: Embed all the things! (Wu et al. (2018))

This researches is stirred up by the idea of word2vec. In the title, Star means "*", which is used to denotes all of things, including words. Actually, this idea has been published in the blog of the skig-gram inventor. This paper is just to abstract the idea and make it be able to apply to other similar tasks. In the following, I will illustrate the idea of this paper by some detailed examples of word embedding.

The Starspace model consists of learning entities, each of which is described by a set of discrete features (bag-of-features. In word embedding model, we call it bag-of-grams) coming from a fixed-length dictionary. For example, for each sentence, words are its features. for each customers, items in his/her shopping records is the features. Previously, we convert words to numerical vectors. Here, we want to convert everything to numerical vectors.

Denoting the dictionary of N features as F which is a N × d embedding matrix. It is very clear when you compare it to the word embedding matrix. After defining the embedding matrix, the researcher embed the entity a with $\sum_{i \in a} F_i$. The definition of entity emebdding can also have other forms, like $\sum_{i \in a} \alpha_i F_i$, where $\sum_i \alpha_i = 1$.

To train this model, the researchers define a loss function:

$$\sum_{(a,b)\in E^+ b^- \in E^-} L^{batch}(sim(a,b), sim(a,b_1^-), ..., sim(a,b_k^-))$$

There are several ingredients to this recipe:

- The generator of positive entity pairs (a, b) coming from the set E^+
- The generator of negative entities b_i^- coming from the set E^-. The researcher utilize a k-negative sampling strategy to select k such negative pairs for each batch update.

- The similarity function $sim(\cdot, \cdot)$ is usually defined as cosine similarity and inner product.
- L_{batch} has two possible functions: margin ranking loss (i.e. $\max(0, \mu - sim(a, b))$, where μ is the margin parameter), and negative log loss of softmax.

9 Researches on Cross-Lingual Mapping between Word Embeddings

This kind of researches is aimed to find a mapping between word embedding of different languages, which is significant in translation tasks.

9.1 A Robust Self-learning Method for Fully Unsupervised Cross-lingual Mappings of Word Embeddings (Artetxe et al. (2018))

Artetxe et al. (2018) proposed a fully unsupervised learning method for learning cross-lingual mappings. In this paper, there is a key and reasonable assumption that two equivalent words in different languages should have a similar distribution. In their proposed method, they want to learn the linear transformation matrices W_X and W_Z, so the mapped embeddings XW_X and XW_Z are in the same cross-lingual space. At same time, they aim to build a dictionary matriX between both languages, which is encoded as a sparse matrix D where $D_{ij} = 1$ if the jth word in the target language is a translation of the ith word in the source language.

9.2 Ideas on Improving Their Proposed Methods

- We can learn a simplex dictionary matrix, rather than a sparse 0-1 matrix.
- The functionality of W_X and W_Z can be learned by employing Deep Neural Networks.

10 Researches on the Performances of Different Models in NLP Tasks

10.1 Baseline Needs More Love: On Simple Word-embedding-based Models and Associated Pooling Mechanisms (Shen et al. (2018))

Because using sophisticated weighting schemes can improve the performance of aggregated word embeddings close to strong LSTM baselines. This, thus, raises the question how much benefit recurrent encoders actually provide over simple word embeddings based methods. The papers illustrated below is aimed to solve this problem.

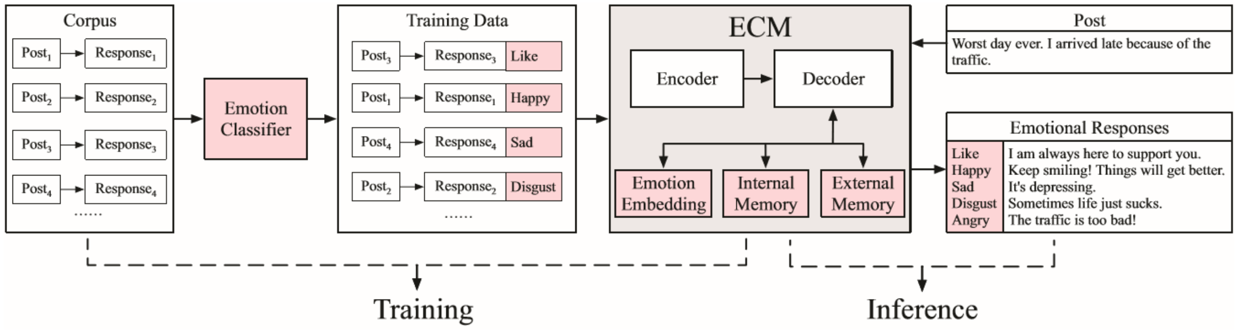
Shen et al. (2018) proposed a new pooling methods, hierarchical pooling, and implemented this new methods into Simple Word-Embedding model. In their experiments, they found their proposed model performs much better than simple RNN and CNN and also costs very little in both computing and time.

All in all, researchers hope to use this paper to emphasize the performance of simple model. In other words, they think the feature engineering is still very important in NLP tasks. If we can construct some efficient features, even the most simple model can perform very well.

11 Researches on ChatBot

11.1 Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory (Zhou et al. (2018))

The researchers addresses the emotion factor in large-scale conversation generation and proposes Emotional Chatting Machine (ECM), which addresses the factor using three new mechanisms that models the high-level abstraction of emotion expressions by embedding emotion categories, captures the change of implicit internal emotion states and uses explicit emotion expressions with external vocabulary.



Post
Worst day ever. I arrived late because of the traffic.

Emotional Responses	
Like	I am always here to support you.
Happy	Keep smiling! Things will get better.
Sad	It's depressing.
Disgust	Sometimes life just sucks.
Angry	The traffic is too bad!

Figure 11: Overview of the whole process: Grey unit (EMC), Pink units: models for emotion factors.

Emotion Embedding

$$s_t = GRU(s_{t-1}, [c_t; e(y_{t-1}); v_e])$$

where v_e is the vector of an emotion category, $e(y_{t-1})$ is the word embedding of y_{t-1} and c_t is the context vector. Based on s_t, the decoding probability distribution can be computed by $softmax(W_o s_t)$ to generate the next token y_t.

Internal Memory

Once the emotion embedding is determined, it would not change. Therefore, the researchers design an internal memory to capture the emotion dynamics during decoding. after getting s_{t-1} from emotion embedding process, the model will enter the internal memory process.

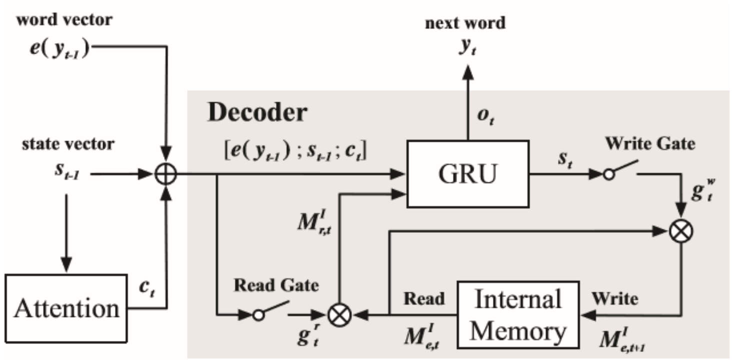


Figure 12: Data flow of the decoder with an internal memory

The whole process are defined as follows:

$$
\begin{aligned}
g_t^r &= sigmoid(W_g^r[e(y_{t-1}; s_{t-1}; c_t)]) \\
g_t^w &= sigmoid(W_g^w s_t) \\
M_{r,t}^I &= g_t^r \otimes M_{e,t}^I \\
M_{e,t+1}^I &= g_t^w \otimes M_{e,t}^I \\
s_t &= GRU(s_{t-1}, [c_t; e(y_{t-1}); M_{r,t}^I])
\end{aligned}
$$

where $\otimes$ is element-wise multiplication, r/w denotes read/write respectively, and I means Internal.

External Memory

Because the relationship between the change of internal emotion state and the selection of a word is implicit and not directly observable, the researcher propose an external memory module to model emotion expression explicitly by assign different generation probabilities to emotion words and non-emotion words. After implementing the decoder with an internal

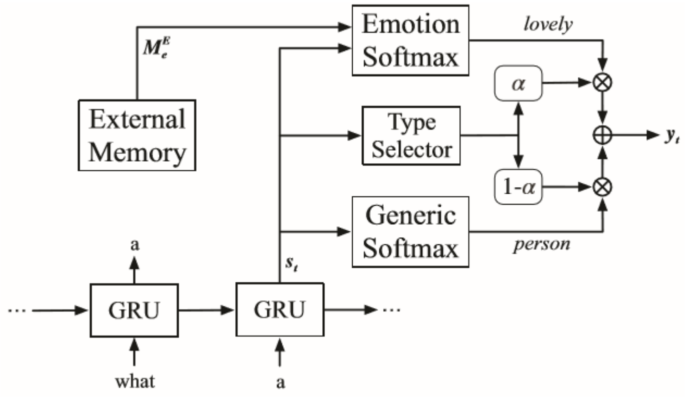


Figure 13: Data flow of the decoder with an external memory

memory, we feed its ouput y_t to decoder with external memory.

$$\begin{aligned} \alpha_t &= sigmoid(v_u^T s_t) \\ P_g(y_t = w_g) &= softmax(W_g^o s_t) \\ P_e(y_t = w_e) &= softmax(W_e^o s_t) \\ y_t \sim o_t P(y_t) &= (1 - \alpha_t) P_g(y_t = w_g) \ or \ \alpha_t P_e(y_t = w_e) \end{aligned}$$

where $P_g(y_t = w_g)$ is generic softmax, $P_e(y_t = w_e)$ is emotion softmax, $\alpha_t \in [0, 1]$ is a scalar to balance the choice between an emotion word w_e and a non-emotion word w_g.

Loss Function

The loss on one sample $< X, Y >$ $(X = x_1, x_2, ..., x_n, Y = y_1, y_2, ..., y_m)$ is defined as:

$$L(\theta) = -\sum_{t=1}^{m} p_t \log(o_t) - \sum_{t=1}^{m} q_t \log(\alpha_t) + ||M_{e,m}^I||$$

where $M_{e,m}^I$ is the internal emotion state at the last step m, α_t is the probability of choosing an emotion word or non-emotion word, and $q_t \in 0, 1$ is the true choice of an emotion word or genetic word in Y. The second term is used to supervise the probability of selecting an emotion or non-emotion word. The third term is used to ensure that the internal emotion state has been expressed completely once the generation is completed.

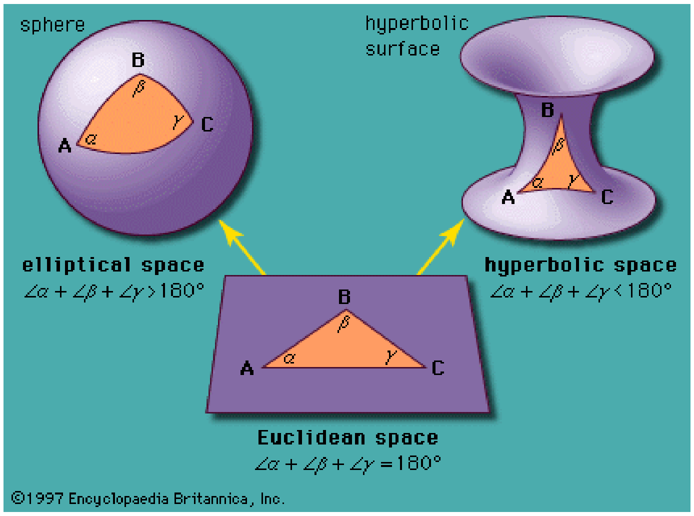
12 Personal Ideas

12.1 Personal Ideas on the Space of Word Embeddings

Recently, a lot of papers focus on embedding text on hyperbolic space (see Figure 14) due to the hierarchy structure. In addition to the language's structure, researchers also start to consider encode the order of words and propose a word2Maxtrix model, because the multiplication of matrix is not commutative. From their work (Mai et al. (2019)), we can clearly summarize that the space of word embedding should have hierarchy structure and be non-commutative and non-associative. In the following, I will illustrate them one by one.

- **Topological space:** some researchers consider the hierarchy structure of the word relations, for example, 'music' should be more abstract than 'musician'. Therefore, they proposed **poincare embeddings** which is defined on hyperbolic space (see Figure 14). This definition is actually a partial order relationship, like the '<', '>' in the natural number set. Thus, to keep this hierarchy property, recent researches take advantage of space structure to encode this kind of hierarchy relationship. Therefore, **we can consider our languages lying in hyperbolic topology space, otherwise, we need to find some methods to encode this kind of structure.**

- **Operator–Communication:** In a latest paper (CMOW, Mai et al. (2019)) , researchers improve CBOW by converting word into matrix, because they find that word order is greatly important to exactly express the meaning of one sentence. For example, if we define a communicative operator *op* to measure the sentences' polarity.


sphere
B
β
A
α
γ
C
hyperbolic surface
B
β
A
α
γ
C
elliptical space
$\angle\alpha + \angle\beta + \angle\gamma > 180°$
hyperbolic space
$\angle\alpha + \angle\beta + \angle\gamma < 180°$
B
β
A
α
γ
C
Euclidean space
$\angle\alpha + \angle\beta + \angle\gamma = 180°$

©1997 Encyclopaedia Britannica, Inc.

Figure 14: Three Different Topological Spaces

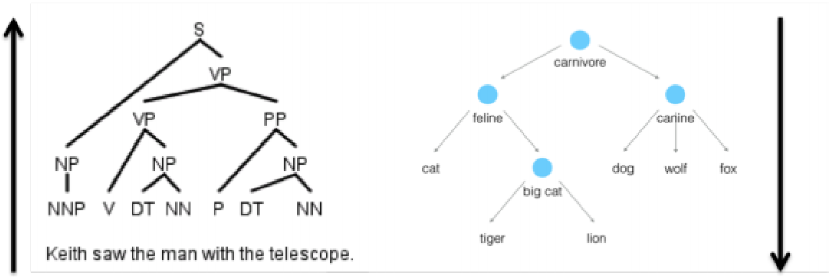


Figure 15: Hierarchy Structures in Language Model

It is obviously that op(”The movie was not awful, it was rather great.”) $\neq$ op(”The movie was not great, it was rather awful.”). Therefore, **the operator should not be commutative.**

- **Operator–Association:** Besides, some researchers proposed that the operator should also be non-associative, i.e. op(”it is not only interesting, but also very comforting.”). It is obviously that op(”not”)$\cdot op$(”only interesting”) $\neq$ op(”not only”)$\cdot op$(”interesting”). Therefore, **the operator should not be associative.**

From my perspective of view, language space where the word embedding should be greatly different from the space we live in, which are typically denoted as Hilbert space. Our language is the abstraction of the complex world and, in order to keep the properties of brief and conciseness, we sacrifice the good properties of commutation and association to build up our languages. After searching for some material online and in the library, I find Lie algebra satisfy the requirements. In the following, I will introduce the definition of Lie algebra.

Definition: A Lie algebra is a vector space g over some field $\mathcal{F}$ together with a binary operator $[\cdot,\cdot] : g \times g \to g$ called the Lie bracket that satisfies the following axioms:

- Bilinearity:

$$[ax + by, z] = a[x, z] + b[y, z]$$
$$[z, ax + by] = a[z, x] + b[z, y]$$

- Alternativity: $[x, x] = 0$
- The Jacobi identity: [x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0
- Anticommutativity (implied): [x, y] = - [y, x]

To sum up, the operator should be non-commutative and non-associative and the embedded space should be unitary-invariant. Recent paper define the operator as the multiplication of matrix and the space as hyperbolic space. In order to further improve this result, we can utilize some non-commutative and non-associative operators, like Lie algebra, and some other topological spaces.

References

Artetxe, M., Labaka, G., and Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 789–798.

Chen, X., Liu, Z., and Sun, M. (2014). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Dhingra, B., Shallue, C. J., Norouzi, M., Dai, A. M., and Dahl, G. E. (2018). Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313.*

Mai, F., Galke, L., and Scherp, A. (2019). Cbow is not all you need: Combining cbow with the compositional matrix space model. *arXiv preprint arXiv:1902.06423.*

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., Li, C., Henao, R., and Carin, L. (2018). Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 440–450.

Tian, F., Dai, H., Bian, J., Gao, B., Zhang, R., Chen, E., and Liu, T.-Y. (2014). A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160.

Wu, L. Y., Fisch, A., Chopra, S., Adams, K., Bordes, A., and Weston, J. (2018). Starspace: Embed all the things! In *Thirty-Second AAAI Conference on Artificial Intelligence.*

Zhou, H., Huang, M., Zhang, T., Zhu, X., and Liu, B. (2018). Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

Appendices

Appendix A: MINUTES

Minutes of the 1st Project Meeting

- Date: 11 Feb. 2019 (Monday)
- Time: 3:30 pm
- Place: Rm. 2541
- Attending: Yansen HAN
- Prof. Nevin Lianwen Zhang
- Absent:None
- Recorder: Yansen HAN

Approval of minutes

- This is first formal group meeting, so there were no minutes to approve.

Discussion Items

Things to do:

1. Preparing a research plan;
2. Reading two assigned papers;
3. Presenting the detailed plan and explaining the key ideas in next meeting.

Meeting adjournment and next meeting

- The meeting was adjourned at 3:50 PM. The next meeting will be held in 1, March.

Minutes of the 2nd Project Meeting

- Date: 1 March 2019 (Monday)
- Time: 3:00 pm
- Place: Rm. 2541
- Attending: Yansen HAN
- Prof. Nevin Lianwen Zhang
- Absent:None
- Recorder: Yansen HAN

Approval of minutes

- This is second formal group meeting.

Discussion Items

Things to do:

1. Time schedule for independent project;
2. Hands-on experiments on NLP.

Meeting adjournment and next meeting

- The meeting was adjourned at 4:50 PM. The next meeting is appointed by ourselves.

Minutes of the 3rd Project Meeting

- Date: 6 May 2019 (Monday)
- Time: 4:00 pm
- Place: Rm. 2541
- Attending: Yansen HAN
- Prof. Nevin Lianwen Zhang
- Absent:None
- Recorder: Yansen HAN

Approval of minutes

- This is third formal group meeting.

Discussion Items

Things to do:

1. make a speech about my project results

Meeting adjournment and next meeting

- The meeting was adjourned at 5:20 PM. The next meeting is May 9, 2019.

Minutes of the 4th Project Meeting

- Date: 8 May 2019 (Wednesday)
- Time: 1:00 pm
- Place: Rm. 2541
- Attending: Yansen HAN
- Prof. Nevin Lianwen Zhang
- Absent:None
- Recorder: Yansen HAN

Approval of minutes

- This is fourth formal group meeting.

Discussion Items

Things to do:

1. submit the report and talk about the potential problems.

Meeting adjournment and next meeting

- The meeting was adjourned at 1:40 PM.