The Third Report of Deep Learning for Natural Language Processing

Yue Zhao
zy2303607@buaa.edu.cn

Abstract

This paper explores the application of Word2Vec models, including Continuous Bag of Words (CBOW) and Skip-Gram, to natural language processing tasks, specifically focusing on the semantic relationships within texts. By encoding words into vector spaces, these models facilitate the mathematical handling of language, allowing for the examination of word semantics based on the proximity and clustering of vectors. The study examines the performance of these models in detecting semantic distances between words and clustering words into meaningful categories, as well as analyzing paragraph similarities within a corpus of novels. Experimental results demonstrate the utility of these models in providing insights into textual data, with Skip-Gram generally outperforming CBOW in tasks involving larger datasets.

Introduction

Vectors: In natural language processing tasks, handing over natural language to algorithms in machine learning typically involves mathematizing language, as machines are not human and only recognize mathematical symbols. Vectors serve as an intermediary through which humans abstract natural language concepts for machine processing, effectively serving as the primary mode of input from humans to machines.

Word embedding: Word embedding, also known as word vectors, is a method for mathematizing words in language, wherein each word is represented as a vector. Prior to training neural networks, words need to be encoded into numerical variables. Word embedding conceptually involves the mathematical embedding of each word from a high-dimensional discrete space to a lower-dimensional continuous vector space. It can also be understood as a mapping process where a word from the text space is mapped or embedded into another numerical vector space through a specific method. This embedding process often involves dimensionality reduction, hence termed as embedding.

Word2Vec: Word2Vec represents an efficient predictive model for learning word embeddings or, equivalently, a language model that generates word vectors. Upon completion of training, the Word2Vec model can map each word to a vector to represent relationships between words, with this vector being the hidden layer of a neural network. The Word2Vec model is based on the bag-of-words assumption, where word order is disregarded.

CBOW & Skip-Gram: Word2Vec primarily comprises the Continuous Bag of Words (CBOW) model and the Skip-Gram model. From an algorithmic perspective, these two approaches are quite similar, differing in that CBOW predicts target words based on source words, or context words, while the Skip-Gram model operates in reverse, predicting context words based on target words. The motivation behind the Skip-Gram model's reversal of the CBOW process lies in CBOW's smoothing of a large amount of distributed information, treating an entire context as a single observation, which can be beneficial for small datasets. In contrast, the Skip-Gram model treats each "context-target word" combination as a new observation, which proves more effective for large datasets.

Methodology

M1: CBOW Model

The CBOW model disregards word order and conveys the entire context using the average vector of its constituent words, subsequently employing this vector for predicting the target word.

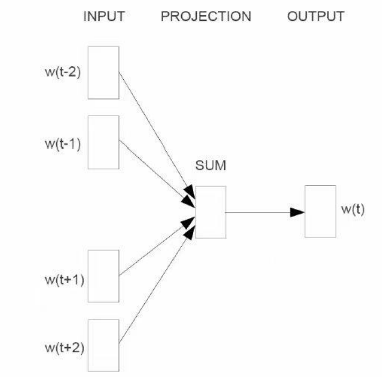

Figure 1：The structure of CBOW model

As depicted in **Figure 1**, the CBOW model aims to maximize the conditional probability of a given sequence of context words to forecast the target word. Given a sequence of context words $(w_{\{1\}}, w_{\{2\}}, ..., w_{\{n\}})$, the CBOW model's objective is to maximize the conditional probability of the target word $w_{\{t\}}$:

$$P(w_{\{t\}} \middle| w_{\{t-k\}}, ..., w_{\{t-1\}}, w_{\{t+1\}}, ..., w_{\{t+k\}}) \tag{1}$$

Where k denotes the context window size.

The CBOW model represents the entire context by averaging the word embedding vectors, subsequently employing the softmax function to compute the conditional probability.

The CBOW model's objective function (loss function) is typically the Negative Log-Likelihood loss, which can be represented as:

$$Loss = -\log P(w_{\{t\}} \middle| w_{\{t-k\}}, ..., w_{\{t-1\}}, w_{\{t+1\}}, ..., w_{\{t+k\}}) \tag{2}$$

M2: Skip-Gram Model

Contrary to CBOW, the Skip-Gram model predicts the probability of context words given a target word.

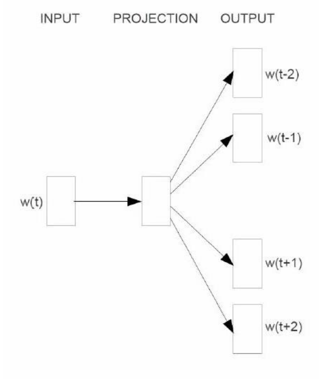

Figure 2: The structure of Skip-Gram model

As depicted in **Figure 2**, the Skip-Gram model aims to maximize the conditional probability of a given target word $w_{\{t\}}$ to predict contextual words. Given the target word $w_{\{t\}}$, the Skip-Gram model seeks to maximize the conditional probability of the sequence of context words $(w_{\{t-k\}},...,w_{\{t-1\}},w_{\{t+1\}},...,w_{\{t+k\}})$:

$$P(w_{\{t-k\}},...,w_{\{t-1\}},w_{\{t+1\}},...,w_{\{t+k\}}\left|w_{\{t\}}\right.) \tag{3}$$

The Skip-Gram model uses the word embedding vector of the target word $w_{\{t\}}$ as input and employs the softmax function to compute the conditional probability.

The objective function (loss function) of the Skip-Gram model is typically the negative log-likelihood loss, which can be represented as:

$$Loss = -\log P(w_{\{t-k\}},...,w_{\{t-1\}},w_{\{t+1\}},...,w_{\{t+k\}}\left|w_{\{t\}}\right.) \tag{4}$$

Experimental Studies

After loading necessary modules, initializing relevant variables, reading novel data, and conducting preprocessing (including filtering and word segmentation), the input data is obtained and then fed into the Word2Vec model for training.

E1: The semantic distance between word vectors

Select four words from each of the 16 input novels, and based on the Word2Vec model training results, use CBOW and Skip Gram models to calculate the semantic distance between the word vectors corresponding to these words. The results are shown in **Table 1**.

Table 1：The semantic distance between related words

Novel names	Related words	Semantic distance	
		CBOW	Skip-Gram
《三十三剑客图.txt》	'范蠡' and '西施'	0.4575192928314209	0.3600348234176636
	'西施' and '王道'	0.28173816204071045	0.5477767586708069
	'王道' and '阿青'	0.35075151920318604	0.5856800079345703
	'阿青' and '范蠡'	0.29348212480545044	0.3335428237915039
《书剑恩仇录.txt》	'乾隆' and '陈家洛'	0.2873528003692627	0.4156913757324219
	'陈家洛' and '霍青桐'	0.2639273405075073	0.3314273357391357
	'霍青桐' and '喀丝丽'	0.5379537045955658	0.5288709998130798
	'喀丝丽' and '乾隆'	0.5169456005096436	0.5917176604270935
《侠客行.txt》	'石破天' and '石中玉'	0.43803417682647705	0.4554518461227417
	'石中玉' and '丁当'	0.4815748929977417	0.4480587840080261
	'丁当' and '白自在'	0.3572646379470825	0.4026216268539429
	'白自在' and '石破天'	0.40220022201538086	0.4747503995895386
《倚天屠龙记.txt》	'张无忌' and '赵敏'	0.28446972370147705	0.3958865404129028
	'赵敏' and '殷素素'	0.17132854461669922	0.3540740609169006
	'殷素素' and '周芷若'	0.15755027532577515	0.3879494667053223
	'周芷若' and '张无忌'	0.23419296741485596	0.3876625299453735
《天龙八部.txt》	'段誉' and '虚竹'	0.2538996934890747	0.4173291921615601
	'虚竹' and '慕容复'	0.2768845558166504	0.4899643063545227
	'慕容复' and '王语嫣'	0.2976822257041931	0.3551492691040039
	'王语嫣' and '段誉'	0.1749885082244873	0.3160059452056885
《射雕英雄传.txt》	'郭靖' and '黄蓉'	0.14131158590316772	0.2753390073776245
	'黄蓉' and '杨康'	0.37305378913879395	0.4728549122810364
	'杨康' and '穆念慈'	0.2800670266151428	0.3651825189590454
	'穆念慈' and '郭靖'	0.31292593479156494	0.5555382370948792
《白马啸西风.txt》	'李文秀' and '苏普'	0.2437152862548828	0.2747992277145386
	'苏普' and '马家骏'	0.5246294140815735	0.4125126600265503
	'马家骏' and '阿曼'	0.48227453231811523	0.3353921771049500
	'阿曼' and '李文秀'	0.23414409160614014	0.2960048913955689
《碧血剑.txt》	'袁承志' and '安小慧'	0.3956354260444641	0.4601105451583862
	'安小慧' and '温青青'	0.4165363311767578	0.3612216711044312
	'温青青' and '阿九'	0.5570563077926636	0.4730565547943115
	'阿九' and '袁承志'	0.44442427158355713	0.4877583384513855
《神雕侠侣.txt》	'杨过' and '小龙女'	0.139786958694458	0.3073266744613648
	'小龙女' and '郭靖'	0.1955062747001648	0.5383422374725342
	'郭靖' and '黄蓉'	0.14131158590316772	0.2753390073776245
	'黄蓉' and '杨过'	0.15716606378555298	0.3968418836593628
《笑傲江湖.txt》	'令狐冲' and '任我行'	0.3180495500564575	0.5262026786804199
	'任我行' and '岳灵珊'	0.45959293842315674	0.6101351678371429
	'岳灵珊' and '东方不败'	0.490595281124115	0.6046977639198303

	'东方不败' and '令狐冲'	0.4730207920074463	0.6013050079345703
《越女剑.txt》	'阿青' and '范蠡'	0.29348212480545044	0.3335428237915039
	'范蠡' and '西施'	0.4575192928314209	0.3600348234176636
	'西施' and '勾践'	0.3105067014694214	0.5167528390884399
	'勾践' and '阿青'	0.2758108377456665	0.5489712357521057
《连城诀.txt》	'狄云' and '戚芳'	0.26116740703582764	0.3544917106628418
	'戚芳' and '戚长发'	0.30007076263427734	0.3525780439376831
	'戚长发' and '丁典'	0.2448725700378418	0.3843483924865723
	'丁典' and '狄云'	0.2510332465171814	0.4004618525505066
《雪山飞狐.txt》	'胡一刀' and '苗人凤'	0.40644484758377075	0.3642376065254211
	'苗人凤' and '苗若兰'	0.38056719303131104	0.4224599599838257
	'苗若兰' and '胡斐'	0.4556869864463806	0.5408146977424622
	'胡斐' and '胡一刀'	0.6690022051334381	0.4878247380256653
《飞狐外传.txt》	'胡斐' and '程灵素'	0.3562692403793335	0.4003747701644898
	'程灵素' and '袁紫衣'	0.3568239212036133	0.4213213920593262
	'袁紫衣' and '苗人凤'	0.2055433988571167	0.4857146739959717
	'苗人凤' and '胡斐'	0.2560257911682129	0.4209439754486084
《鸳鸯刀.txt》	'林玉龙' and '任飞燕'	0.09004473686218262	0.1254056692123413
	'任飞燕' and '常长风'	0.26044315099716187	0.2111302614212036
	'常长风' and '逍遥子'	0.2084333896636963	0.2799006104469299
	'逍遥子' and '林玉龙'	0.21568691730499268	0.2826634645462036
《鹿鼎记.txt》	'康熙' and '韦小宝'	0.32289958000183105	0.3858333826065064
	'韦小宝' and '陈近南'	0.5459137558937073	0.5850825309753418
	'陈近南' and '方怡'	0.54404217004776	0.5760278701782227
	'方怡' and '康熙'	0.5965461432933807	0.7904239892959595

According to **Table 1**, on the one hand, in the same novel, the semantic distance between the main characters is very small, with the vast majority below 0.5. For example, in the novel "《鸳鸯刀.txt》", the semantic distance calculated by the CBOW model between "林玉龙" and "任飞燕" is as low as 0.09, indicating a deep correlation between the two. And only a few semantic distances exceed 0.5, which may be due to insufficient model training or its own limitations.

On the other hand, compared with the CBOW model, the Skip-Gram model obtains larger semantic distances between most of the main characters, indicating that the model performs worse than the CBOW model within the same novel or in the local range of the input dataset.

E2: Clustering of a certain category of words

Find three types of word sets extracted from raw novel data online, namely people, gang, and kungfu. For these three types of datasets, 10 words belonging to the trained model are randomly sampled and combined as clustering objects. Based on the training results of the Word2Vec model, the word vectors corresponding to the sampled words under CBOW and Skip Gram methods can be obtained, respectively. Finally, the KMeans method is used to cluster them, and the results are shown in **Table 2**:

Table 2：The KMeans Cluster results

Real category	KMeans Cluster results			
	CBOW		Skip-Gram	
people	'丁大全'	0	'高老者'	0
	'曾柔'	0	'梅文馨'	0
	'凯别兴'	0	'林朝英'	1
	'宫九佳'	0	'郭破虏'	0
	'易大彪'	0	'林玉龙'	0
	'平阿四'	0	'钟志灵'	0
	'胡青牛'	0	'邓百川'	0
	'萧远山'	0	'田青文'	0
	'唐六爷'	0	'胡大哥'	0
	'喀丝丽'	0	'梁自进'	0
gang	'南少林'	0	'桃花岛'	2
	'巨鲸帮'	0	'玄素庄'	2
	'血刀门'	0	'清凉寺'	2
	'青城派'	1	'言家拳'	0
	'绝情谷'	0	'金刀寨'	0
	'神龙教'	0	'绝情谷'	2
	'铁剑门'	0	'明教'	2
	'桃花岛'	2	'武当派'	1
	'日月神教'	0	'巨鲸帮'	0
	'逍遥派'	0	'古墓派'	1
kangfu	'太极拳'	1	'狮子吼'	1
	'火焰刀'	0	'蛤蟆功'	1
	'易筋经'	1	'沐家拳'	0
	'八卦刀'	0	'梅花拳'	0
	'伏虎掌'	0	'六合拳'	1
	'黑沙掌'	0	'韦陀掌'	1
	'打狗棒法'	1	'独孤九剑'	1
	'查拳'	0	'太极剑'	1
	'一阳指'	1	'破玉拳'	1
	'混元功'	0	'袖里乾坤'	0

From **Table 2**, it can be seen that in the CBOW model, all "people" words are clustered to "0"; At the same time, in the Skip Gram model, only one "people" class word was not clustered to "0", indicating that both models are very accurate in clustering "people" class words.

For words like "gang", under the Skip Gram model, half of them are clustered to "2"; Under the CBOW model, 80% of them are clustered to the same number "0" as the "people" class. Similarly, for words like "kangfu", under the Skip Gram model, 70% of them are clustered to "1"; Under the CBOW model, 60% of them are still clustered to the same number "0" as the "people" class, and only 40% are clustered to "1". This indicates that, on the one hand, the clustering performance of both models on words such as "gang" and "kangfu" is worse than that of "people"; On the other hand, overall, the Skip Gram model performs better in clustering than the CBOW model.

E3: The semantic association of paragraphs

Paragraph sampling is performed from the preprocessed input dataset, with two samples taken from each of the 16 novels, for a total of 32 sampled paragraphs. Based on the training results of the Word2Vec model, the set of word vectors corresponding to each word in the sampled paragraphs under CBOW and Skip Gram methods can be obtained. Finally, using dot product operation to obtain the similarity between these 32 sampled paragraphs, the results are shown in **Table 3**.

Table 3：The Similarity calculation results

Paragraph		Real similarity	Similarity calculation results	
			CBOW	Skip-Gram
0	1	1	0.2381952852010727	0.8764129281044006
1	2	0	0.3658696115016937	0.5029076337814331
2	3	1	0.7996270656585693	0.9535430073738098
3	4	0	0.5433640480041504	0.8651210069656372
4	5	1	0.666031539440155	0.838506281375885
5	6	0	0.4922163486480713	0.6676362752914429
6	7	1	0.29072973132133484	0.6926386952400208
7	8	0	0.5900549292564392	0.7443312406539917
8	9	1	0.4993681013584137	0.6906418204307556
9	10	0	0.47298213839530945	0.49442583322525024
10	11	1	0.4462793171405792	0.5969835519790649
11	12	0	0.2651570439338684	0.6629307270050049
12	13	1	0.31963056325912476	0.7680500149726868
13	14	0	0.35884276032447815	0.7182556986808777
14	15	1	0.6038172245025635	0.7378352284431458
15	16	0	0.8571489453315735	0.7695199251174927
16	17	1	0.3235613703727722	0.7680907845497131
17	18	0	0.6048038005828857	0.9213566184043884
18	19	1	0.7339557409286499	0.904191255569458
19	20	0	0.6329721808433533	0.8430131077766418
20	21	1	0.4338639974594116	0.5285816192626953
21	22	0	0.19748526811599731	0.6108472347259521
22	23	1	0.6848456859588623	0.7779313325881958
23	24	0	0.38947781920433044	0.8619401454925537
24	25	1	0.6885862946510315	0.8336378335952759
25	26	0	0.7970162630081177	0.845542848110199
26	27	1	0.4639354944229126	0.9733167290687561
27	28	0	0.38719630241394043	0.8334564566612244
28	29	1	0.5591506958007812	0.5767935514450073
29	30	0	0.631257176399231	0.7310351133346558
30	31	1	0.5886297821998596	0.7819842100143433
31	0	0	0.2381952852010727	0.8764129281044006

From **Table 3**, it can be seen that overall, the accuracy of both the CBOW model and Skip Gram model is around 53% (bounded by 0.5). In addition, the overall similarity calculated under the Skip Gram model is relatively high, indicating that it performs better in similarity judgment between paragraphs with a true similarity of 1, that is, between paragraphs belonging to the same novel. Under the CBOW model, whether the true similarity is 0 or 1, the similarity judgment effect does not differ significantly.

Conclusions

The findings from the application of Word2Vec models to natural language processing illustrate their effectiveness in capturing the semantic properties of words and paragraphs. Both CBOW and Skip-Gram models successfully identified semantic distances and grouped similar words through clustering techniques. However, the Skip-Gram model displayed superior performance, particularly in handling larger datasets and more complex semantic relationships. This superiority is evident in tasks such as semantic distance measurement and paragraph similarity assessment, where Skip-Gram achieved more distinct and accurate clustering of words and maintained higher similarity scores between paragraphs of the same context. The study underscores the potential of embedding models to enhance the understanding and processing of natural language, suggesting avenues for future research in refining these models for broader linguistic applications.

References

[1]https://blog.csdn.net/ARPOSPF/article/details/128397083?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171751130316800178529050%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171751130316800178529050&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-1-128397083-null-null.142^v100^pc_search_result_base2&utm_term=cbow%E5%92%8Cskipgram%20%E6%A8%A1%E5%9E%8B&spm=1018.2226.3001.4187