Neural Language Model - Processamento de linguagem natural

Pedro V. Brum

[1]Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

pedrobrum@dcc.ufmg.br

1. Introdução

Nesse trabalho foi consruído um modelo de linguagem utilizando os dados disponibilizados, que consistem de texto coletado online. Esses dados foram utilizados como corpus para o modelo de linguagem a ser criado. A partir do modelo de linguagem, o objetivo é avaliar o quão bem ele funciona para a tarefa de analogia. Ou seja, analisar se os *embeddings* formados pelo modelo construído conseguem representar as palavras do corpus de forma eficaz. Antes de tudo, foi feito um preprocessamento do texto, retirando pontuações e $stopwords$ e colocando todas as letras do texto como minúsculas. Com o texto preprocessado, um modelo de linguagem é criado utilizando $Word2Vec$. Para implementar o trabalho foi utilizada a linguagem python e os pacotes $Gensim$ e $NLTK$.

2. Parâmetros

O algoritmo utilizado na execução desse trabalho possui alguns parâmetros que podem ser estudados como: número de iterações utilizados para treinar o modelo, tamanho da janela de texto, tamanho do *embeddings* a serem formados, método utilizado para formar os *embeddings* (CBOW ou Skip-gram), tamanho do texto utilizado para treino, entre outros.

3. Análise experimental

Os parâmetros escolhidos para realizar a análise experimental foram: tamanho da janela, tamanho do texto utilizado para treinar o modelom o método utilizado para formar os *embeddings*, tamanho dos *embeddings* formados e número de iterações para treinar o modelo. Em relação ao tamanho do texto utilizado para treino, foram consideradas o número de palavras do texto.

O corpus original, proveniente do preprocessamento do texto disponível, possui 10,890,639 palavras no total, incluindo palavras repetidas. Foram gerados outros corpus utilizando 90%, 80%, 70%, 60% e 50% das palavras do corpus original.

Para cada combinação de parâmetros foram construídos modelos de linguagem utilizando os corpus formados. Foi feita uma análise de perfomance desses modelos a partir de analogias, compostas por 4 palavras. Assim, dado um conjunto de 3 palavras, o modelo de linguagem deve ser capaz de prever a palavra mais próxima desse conjunto. No arquivo disponibilizado ($question - words.txt$) existem 19.544 analogias. Não necessariamente, os corpus gerados possuem todas as palavras que formam as analogias. As analogias que possuem pelo menos uma palavra que não está no corpus foram ignoradas. A métrica escolhida para comparar as perfomaces dos modelos foi a média da distância de similaridade entre a palavra alvo e a palavra predita pelo modelo para todas as analogias.

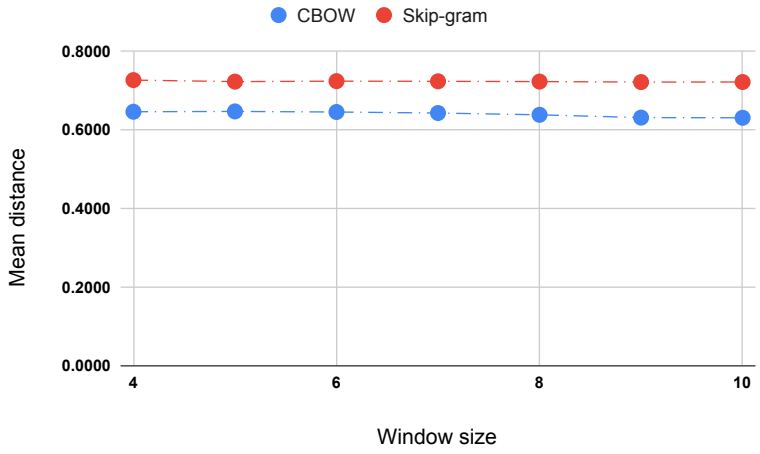

Figure 1. Variação do tamanho da janela.

Table 1. Variação do tamanho do corpus.

	Method					
	CBOW			*Skip-gram*		
Training size	analog	errors	mean distance	analog	errors	mean distance
50%	18,122	1,422	0.5930	18,122	1,422	0.6719
60%	18,401	1,143	0.5988	18,401	1,143	0.6790
70%	18,476	1,068	0.6057	18,476	1,068	0.6918
80%	18,532	1,012	0.6175	18,532	1,012	0.7011
90%	18,784	760	0.6300	18,784	760	0.7171
100%	18,900	644	0.6315	18,900	644	0.7225

Para observar o efeito da variação do tamanho da janela no desempenho do modelo, foi fixado o tamanho do *embedding* igual 100 e 10 iterações. Além disso, nesse caso foi utilizado o corpus completo, com 10,890,639 palavras. A figura 1 apresenta como a média da distância de similaridade varia com o tamanho da janela utilizada para treinar o modelo. É evidente que o tamanho janela impacta pouco no valor da média. As médias da distância quando utilizamos tamanho igual a 4 e quando utilizamos tamanho igual a 10 são bem próximas, tanto quando utilizamos o *CBOW* como quando utilizamos o *Skip-gram*. A partir disso, podemos afirmar que o número de palavras usadas para formar o contexto de uma palavra não impacta o modelo de forma significativa, utilizando o conjunto de dados disponível.

Para observar o efeito da variação do tamanho do corpus no desempenho do modelo, foi fixado o tamanho do *embedding* igual 100, 10 iterações e tamanho de janela igual a 10. A figura 2 e a tabela 3 apresentam como a média da distância de similaridade varia com o tamanho do corpus utilizado para treinar o modelo. Pode-se observar que quanto maior o tamanho do corpus maior é a média da distância de similaridade, tanto quando utilizamos o *CBOW* como quando utilizamos o *Skip-gram*. Porém, não podemos afirmar que os modelos de linguagem formados com corpus menores que o origianl são melhores. Quando utilizamos corpus menores, o vocabulário diminui. Ou seja, o número de palavras únicas diminui. Dessa forma, o número de analogias que possuem pelo menos uma palavra que não está no vocabulário aumenta, e portanto, é natural que a média da

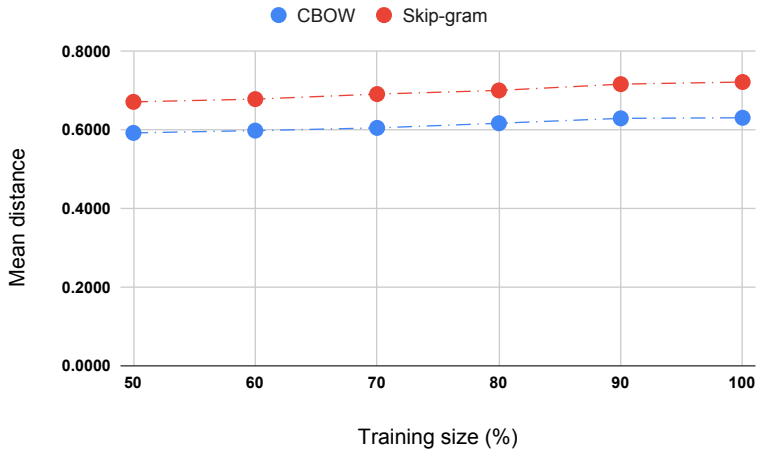

Figure 2. Variação do tamanho do corpus.

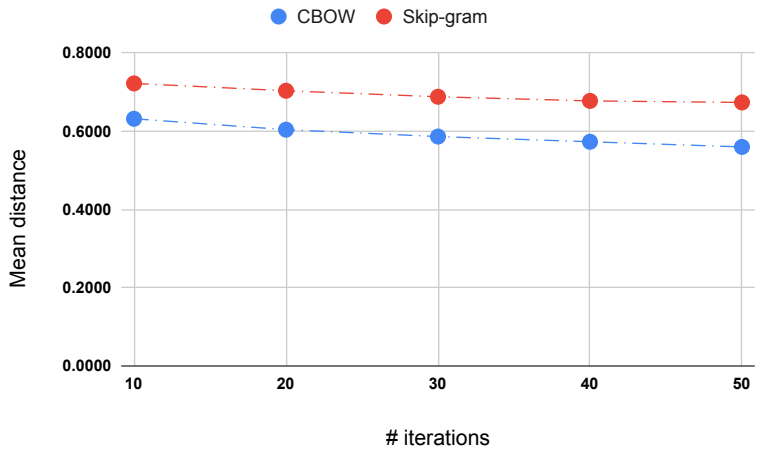

Figure 3. Variação do número de iterações utilizadas para treinar o modelo.

distância diminua. Quanto menor o número de analogias consideradas maior tende ser a média.

Para observar o efeito da variação do número de iterações no desempenho do modelo, foi fixado o tamanho do *embedding* igual 100 e tamanho de janela igual a 10. Além disso, nesse caso foi utilizado o corpus completo. A figura 3 apresenta como a média da distância de similaridade varia com o número de iterações utilizadas para treinar o modelo. Quanto maior o número de iterações menor tende ser a média das distâncias, tanto quando utilizamos o *CBOW* como quando utilizamos o *Skip-gram*. Porém, quanto maior o número de iterações maior é o tempo gasto para treinar o modelo, principalmente quando utilizamos o método *Skip-gram*.

Para observar o efeito da variação do tamanho dos *embeddings* formados no desempenho do modelo, foi fixado o tamanho de janela igual a 10 e 10 iterações. Além disso, nesse caso foi utilizado o corpus completo. A figura 4 apresenta como a média da distância de similaridade varia com o tamanho dos *embeddings*. Quanto maior tamanho

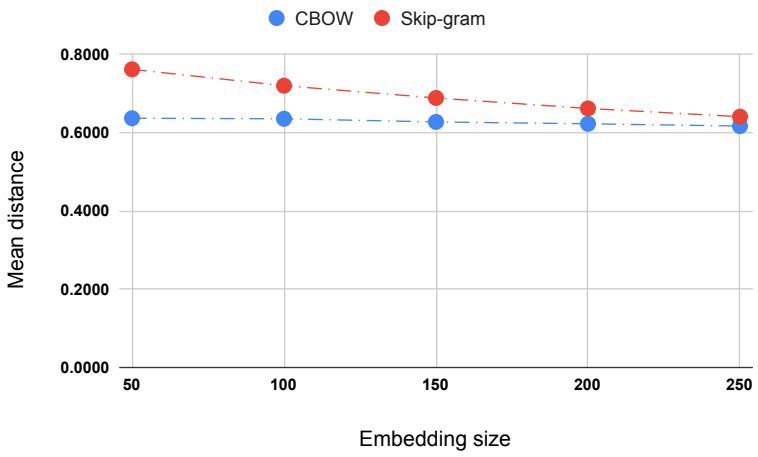

Figure 4. Variação do tamanho dos *embeddings* formados durante o treinamento.

dos *embeddings* menor tende ser a média das distâncias. De forma análoga ao que ocorre com o número de iterações, quanto maior o tamanho dos *embeddings* maior é o tempo gasto para treinar o modelo, principalmente quando utilizamos o método *Skip-gram*. Nesse caso, podemos observar que o método *Skip-gram* consegue obter resultados próximos a quando utilizamos o método *CBOW*, quando utilizamos tamanho de *embedding* igual a 250. Além disso, temos que utilizando o *Skip-gram* a variação da média é maior conforme variamos o tamanhos do *embeddings*.

Podemos notar que o método *CBOW* consegue obter resultados melhores em relação ao método *Skip-gram* no geral. Isso não significa que o *CBOW* é sempre melhor que o *Skip-gram*. O *CBOW* é melhor nesse caso, utilizando o corpus e as analogias disponíveis. Além disso, foi possível observar que o método *Skip-gram* demora mais para executar.

A partir disso, utilizando tamanho de janela igual a 20, tamanho de *embeddings* igual a 250, 100 iterações, o corpus original e o método *CBOW*, foi possível obter um modelo cuja a média das distâncias é igual a **0.36273**.

4. Conclusão

Neste trabalho foram construído modelos de linguagem para tarefa de analogia. A partir do estudo de parâmetros, podemos observar que os melhor resultados são obtidos quando utilizamos o método *CBOW*. Além disso, foi possível observar que o método *Skip-gram* é mais lento que o método *CBOW* e que quanto maior o número de iterações e o tamanho dos *embeddings* melhor tende ser o resultados finais. Ainda em relação aos parâmetros estudados, temos que o tamanho da janela não impacta tanto os resultados em relação aos outro parâmetros.