Disciplinas: Algoritmos de Reconhecimento de Padrões e Redes Neurais Profundas

DeepText: usando semântica de contexto para auto-completar palavras e gerar texto automaticamente

Altino Dantas

Introdução

As sentenças de uma linguagem escrita são produzidas pela colocação sequencial de elementos próprios do seu vocabulário, de forma que a combinação gere a mensagem desejada. Na escrita, a inserção de um elemento possui relação com os termos que o antecedem.

Por esse princípio, processadores de texto e IDEs, costumam oferecer mecanismo para auto-completar palavras ou sentenças. Todavia, tipicamente, este recurso apresenta tão somente uma lista com opções de palavras cujos caracteres iniciais coincidem com os do termo corrente, ignorando assim um contexto mais amplo.

Este trabalho utiliza uma rede neural recorrente que, treinada com um conjunto de sequências de caracteres de uma determinada linguagem, obtém a capacidade de escrever automaticamente palavras ou blocos de texto a partir de uma entrada do contexto treinado.

Materiais e métodos

Modelo LSTM (Long short-term memory)

A Figura 1 apresenta a arquitetura da rede neural implementada no framework Keras [1]. A camada *Fully Connected* foi configurada com entrada igual à quantidade de elementos do vocabulário do corpus.

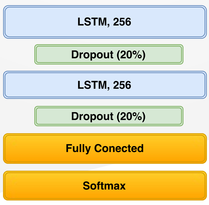

Figura 1: Arquitetura de rede LSTM utilizada no trabalho.

Construção dos *corpus*

Foram pré-processados textos de três diferentes domínios a fim de que se construísse um *corpus* de treinamento para cada um deles. A Tabela 1 mostra a quantidade de caracteres únicos, linhas e total caracteres considerando todas as sentenças em cada um dos três conjuntos.

Tabela 1: *Corpus* utilizados no estudo

Corpus	Caracteres únicos	Total de linha	Total de caracteres
SBSEThesis	88	2.311	771.179
Bible	63	32.359	3.924.374
JavaCode	69	436.565	12.053.424

- **SBSEThesis** possui sentenças no idioma Português e foi composto pela compilação de todos os textos de sete dissertações de mestrado no tema SBSE. Foram removidos elementos não textuais, tais como figuras.
- **Bible** foi constituído por todos os textos dos livros da bíblia, considerando o antigo e novo testamentos ambos em língua portuguesa.
- **JavaCode** foi composto pelos arquivos Java de 20 versões do projeto Lang, obtidas com o Defects4J [2]. Foram removidos todos os tipos de comentários e espaços de tabulação.

Treinamento e avaliação

Para cada um dos *corpus* disponíveis a rede foi treinada por 60 épocas, com o algoritmo RMSProp (taxa de aprendizado = 0,01) para cálculo de custo e considerando-se 40 caracteres como na janela de recorrência.

Para avaliar a capacidade do modelo em predizer corretamente o complemento de uma sequência, foram aleatoriamente selecionadas 100 sequências de cada *corpus* e então removida a parte a ser predita.

Resultados

A Figura 2 apresenta o percentual de palavras (ou sequências) auto-completadas corretamente para cada *corpus*

Figura 2: Taxa de sucesso de predição por *corpus*.

No contexto da linguagem de programação Java, o modelo conseguiu auto-completar corretamente 56% das sequências geradas. Possivelmente, este resultado pode ser melhorado ampliando-se as dimensões da rede neural para melhor acomodar o tamanho do deste *corpus*. Nos dois domínio de aplicação com o idioma português, o sucesso foi de 60% e 70%. Além disso, apenas 1 das 200 palavras preditas não pertencia ao idioma, demonstrando assim capacidade em "aprender" corretamente um padrão de escrita.

INDO ALÉM... GERANDO BLOCOS DE TEXTO

ENTRADA	**josias que esta páscoa foi celebrada ao senhor em jer**
SAÍDA MODELO	*usalém e a sua casa de israel, e a terra de todos os filhos de israel, e disse: eis que o senhor se achará de ti a minha mãe e o seu povo de deus de todos os seus pais, e a terra de arão e a tua mãe de meu pai e a sua casa de judá*
ENTRADA	**inicialmente é mostrada, na seção 4.1, uma visão gera**
SAÍDA MODELO	*l, o algoritmo de busca e a segunda instância e a avaliação de um problema de otimização em sbse possui um conjunto de soluções de sbse e a avaliação de custo de execução.*
ENTRADA	**public boolean addregex(fastDateParse parse, stringBuilder reg**
SAÍDA MODELO	*ex, String str, String separator) { if (*

Considerações finais

A rede LSTM se mostrou promissora para a tarefa de auto-completar sentenças independente do domínio da linguagem utilizada. Os resultados obtidos indicam de que a abordagem pode ser utilizada para sintetização de blocos de textos, inclusive códigos fonte em linguagem de programação.

Referências

[1] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[2] R. Just, D. Jalali, and M. D. Ernst. Defects4j: A database of existing faults to enable controlled testing studies for java programs. In Proceedings of the 2014 International Symposium on Software Testing and Analysis, ISSTA 2014, pages 437–440, New York, NY, USA, 2014. ACM.

Professores orientadores: Anderson da Silva Soares, Celso Camilo, Gustavo Teodoro Laureano