UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Instituto de Física
Bacharelado em Engenharia Física
Tópicos especiais de engenharia física

Deomar Santos da Silva Junior / 00260682

Geração de sentenças baseadas no vocabulário e tendência de uso de palavras do humorista *Hasan Minhaj* utilizando cadeias de Markov

Introdução

Cadeias de Markov são modelos estocásticos que descrevem uma sequências de estados que dependem apenas do estado anterior. Por isso, cadeias de markov são modelos possíveis de serem aplicados para simular os comportamentos humanos que, muitas vezes, baseados no estado atual, decidem uma ação que define o estado futuro como, por exemplo, dirigir um carro. Quando você está no trânsito, a posição dos carros e velocidades no tempo atual, definem a direção e a velocidade que você tomará no tempo futuro. Outras aplicações envolvem a geração de músicas, reconhecimento de texto e geração de sentenças que pode ser modelado como uma cadeia de palavras, que são os estados, em que a próxima palavra é definida pela palavra anterior. Neste trabalho, o modelo de cadeias de markov será aplicado para a geração de sentenças a partir da transcrição de um stand-up do humorista Hasan Minhaj [1].

Cadeias de markov para aplicação em linguagem de processamento natural (NLP)

Para a aplicação do modelo em NLP, considera-se que cada palavra representa um estado da cadeia de markov. Os possíveis próximos estados da cadeia, estados ***j***, são o conjunto de palavras que são utilizadas após a palavra atual, estado ***i***. O estado futuro da cadeia será determinado pela probabilidade de transição do estado ***i*** para o estado ***j***. Sendo, por exemplo, o estado atual da cadeia a palavra *'has'*, os possíveis próximos estados são 'a', 'to', 'been', 'this', 'taught', 'ever', 'all' e 'passed' (figura 1).

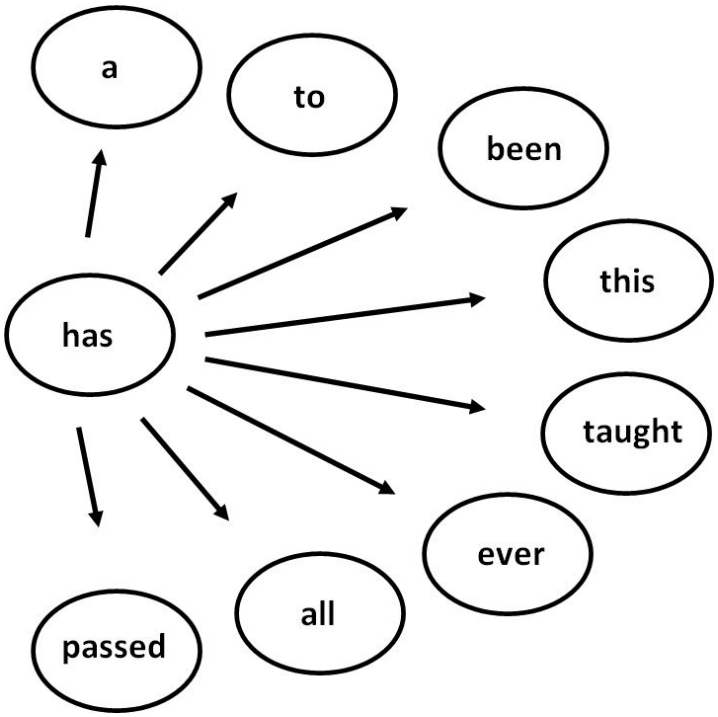

Figura 1. Cadeia de markov aplicado em NLP.

Para criar os possíveis próximos estados, são selecionadas todas as palavras que, em determinado contexto, foram utilizadas após a palavra atual. Depois, atribui-se uma probabilidade de transição do estado ***i*** para o estado ***j*** que é determinada pela frequência de uso de cada possível estado. Por exemplo, caso a palavra 'all' apareça em 20% das vezes depois da palavra 'has', a probabilidade de transição do estado 'has' para o estado 'all' é de 20%. Para o presente trabalho, foi utilizada a transcrição do stand-up do humorista *Hasan Minhaj* intitulado *'Homecoming King'* [2].

Probabilidade de transição

Primeiramente, é criado um dicionário em que as *chaves* do dicionário são todas as palavras da transcrição e, como *valores*, todas as palavras que já foram utilizadas após a palavra da *chave*. Então, conta-se a frequência com que cada palavra é utilizada e monta-se o vetor de transição v do estado ***i*** para o estado ***j***.

Para o exemplo da figura 1, tem-se as seguintes probabilidades (figura 2):

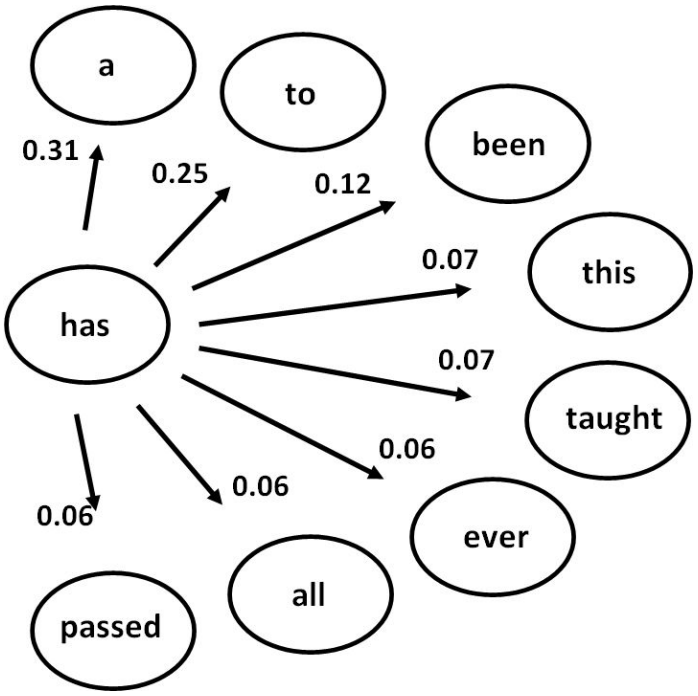

Figura 2. Probabilidades de transição da cadeia de markov.

De forma que o vetor v dado por:

$$v = \begin{bmatrix} 0.31 \\ 0.25 \\ 0.12 \\ 0.07 \\ 0.07 \\ 0.06 \\ 0.06 \\ 0.06 \end{bmatrix}$$

Em que v representa a probabilidade de transição do estado da palavra 'has' para, de cima para baixo, 'a', 'to', 'been', 'this', 'taught', 'ever', 'all' e 'passed'. Diferente da cadeia de markov usual, há um vetor de transição diferente para cada estado da cadeia. Também não é necessária a criação da matriz de transição uma vez que o estado futuro é determinado apenas pelo vetor de transição.

Construção das sentenças

Para a construção da sentença, calcula-se o vetor de transição da palavra do estado ***i*** para o estado ***j*** e, então, faz-se a transição para uma palavra dos possíveis estados futuros de forma estocástica

em que a probabilidade de transição é dada pelo vetor v. Para os estados seguintes o processo se repete de forma que o vetor é calculado para cada palavra da sentença.

Após rodar o programa em *python* escolhendo-se aleatoriamente a primeira palavra da sentença, obteve-se do modelo:

Yeah, what the airport. We got to say.

A sentença acima não está incluída na transcrição original, concluindo que se trata de uma nova formação. E pode ser interpretada como uma reação positiva de duas ou mais pessoas ao verem um aeroporto. A tradução da sentença seria algo como: "Yeah. Que aeroporto! Nós temos que falar/admitir."

Outros resultados do modelo são:

I had said, "Describe your dreams." Then he says it down.

I told me!

Como pode-se perceber, nem sempre as frases respeitam a gramática ou tem algum sentido, como no caso "I told me" que está errada gramaticalmente. Provavelmente a palavra "told" foi escrita após "I", pois é uma construção frequentemente utilizada no texto original. Após uma breve busca, encontra-se, entre outras, no texto original:

I told you

I told my mom

Justificando a escolha da palavra "told" após "I". O mesmo ocorre para a palavra "me". Algumas ocorrências no texto:

Nobody told me!

But no one told me about it.

A construção correta da sentença seria "I told myself".

Tempo de primeira passagem

Após pesquisa feita na área de NLP por modelos de cadeias de markov, não foi possível encontrar a aplicação do conceito de tempo de primeira passagem. Uma explicação pode ser o fato de que dado um estado ***i***, pode não ser possível chegar num estado ***j***, pois as palavras podem não ter um "caminho de ligação" ou podem ter um caminho infinito dentro dos infinitos estados possíveis dentro de uma sentença.

Conclusões

As sentenças geradas pelo modelo respeitam, em geral, a construção básica de sentenças compostas por sujeito, verbo e objeto. De forma que pode ser considerado um sucesso dado o modelo básico utilizado. Para gerar sentenças mais precisas, pode-se controlar outros fatores

como a posição da frase (inicio, meio, fim), a frequência de palavras na mesma frase e, também, pode-se filtrar grupos de palavras para serem utilizadas. Por exemplo, pode-se filtrar apenas o uso de X palavras que são do grupo “geografia” para a geração de sentenças relacionadas ao assunto evitando assim a fuga do assunto ao longo da sentença.

Referências:

[1] Netflix - Hasan Minhaj: Homecoming King **<Disponível em 14/06/2020>**

[2] HASAN MINHAJ: HOMECOMING KING (2017) - Full Transcript **<Disponível em 14/06/2020>**