Universidade Federal de Minas Gerais
Departamento de Ciência da Computação

POS tagging

Rafael Torres Souza.
rafaeltsouza@dcc.ufmg.br

25 de Novembro de 2018

Resumo

O propósito do trabalho foi implementar e avaliar um modelo de predição de POS tagging utilizando uma rede neural LSTM, com o intuito de analisar a precisão das predições, analisando especificamente também a precisão de acordo com a classe da palavra.

1 Introdução

A tarefa de POS Tagging resumidamente consiste em classificar palavras em classes gramaticais, como verbo, sujeito, préposição, etc. Para tal tarefa foi implementada uma rede neural recorrente, uma LSTM. A rede neural foi treinada utilizando a base de treinamento do corpus Mac-Morpho, assim como para avaliar a rede também foi utilizada a base de teste do mesmo corpus.

A escolha pela LSTM (*Long short-term memory*) se justifica na sua capacidade de utilizar o contexto anterior para predizer a próxima palavra; em resumo, sua capacidade de se "lembrar"do que veio antes da palavra alvo. Redes neurais recorrentes são compostas por céluas que recebem o input atual, e após passar pela função de ativação "esquecem"parte do dado e guardam outra parte com o intuito de passar esse dado para a célula seguinte, que irá utilizar esse dado junto do input atual, formando uma rede de células que leva em conta o contexto.

Após implementar a rede, o trabalho consiste em avaliar questões como o melhor tamanho de janela para predição, e avaliar a precisão da predição não só para qualquer classe possível no corpus, mas também avaliar essa precisão para cada classe em específico.

2 Implementação

O trabalho foi desenvolvido utilizando a linguagem *Python 3*. Para criar a base de treinamento à partir do corpus, primeiro o conjunto de todas as classes da base de treinamento (na mesma ordem do arquivo do corpus) foi convertida para uma representação *one-hot*; Tendo essa representação, foi utilizado um conceito de "janela-deslizante"para separar os dados em dois grupos: sequência de classes gramáticas de uma frase no primeiro grupo, e no segundo grupo a classe gramátical da palavra que viria logo após a sequência do primeiro grupo.

Para ficar mais tangível: Supondo que tenhamos uma frase com a seguinte sequência de classes gramáticas: artigo, nome, verbo, preposição e nome. Para uma janela de tamanho igual à 4 (quatro) o código irá separar no primeiro grupo as representações *one-hot* das quatro primeiras classes, e no segundo grupo a representação *one-hot* da quinta classe gramátical. Imaginando que o corpus tem mais que cinco palavras, após essa primeira iteração seria feito a mesma coisa na segunda iteração, mas colocando no primeiro grupo as representações das classes da segunda, terceira, quarta e quinta palavras, enquanto no segundo grupo estaria a representação da classe da sexta palavra. Essa iterações continuariam até o momento em que a representação da classe da

última palavra do corpus é inserida no segundo grupo. Basicamente isso irá gerar duas coleções de mesmo tamanho, onde em cada posição da primeira coleção está uma sequência de classes do tamanho da janela definida, e na mesma posição na outra coleção está a classe que viria em sequência.

Com essas duas coleções a rede é treinada. A rede foi montanda com uma camada de entrada contendo 50 neurônios e utilizando *relu* e *sigmoid* como funções de ativação em seguida. O treinamento da rede para os resultados que serão apresentados foi com 10 (dez) épocas.

Com a rede já treinada, foi criada a base de teste com a mesma técnica utilizada para criar a base de treinamento, com uma diferença: além de uma base organizada da mesma forma que a base de treino, também foi criada uma base onde os grupos foram divididos de acordo com a classe que deveria ser retornada pela rede. Ou seja, se ao preparar a base de treinamento se obtiveram 2 (duas) coleções, cada uma contendo representações de classes, ao preparar a base de testes se obtiveram 52 (cinquenta e duas) coleções, sendo que para cada uma das 26 (vinte e seis) classes gramáticas presentes no corpus, existiam 2 (coleções) contendo representações de classes. Com isso foi possível saber não só a precisão da rede de uma forma "agnóstica"à classe, mas também saber a precisão do modelo para predizer cada classe do corpus.

3 Análise Experimental

Um primeiro experimento foi analisar a precisão da rede independente da classe da palavra alvo, variando o tamanho da janela. Foram feitos testes com a janela de tamanho igual à 3 (três), até com a janela de tamanho igual à 10 (dez).

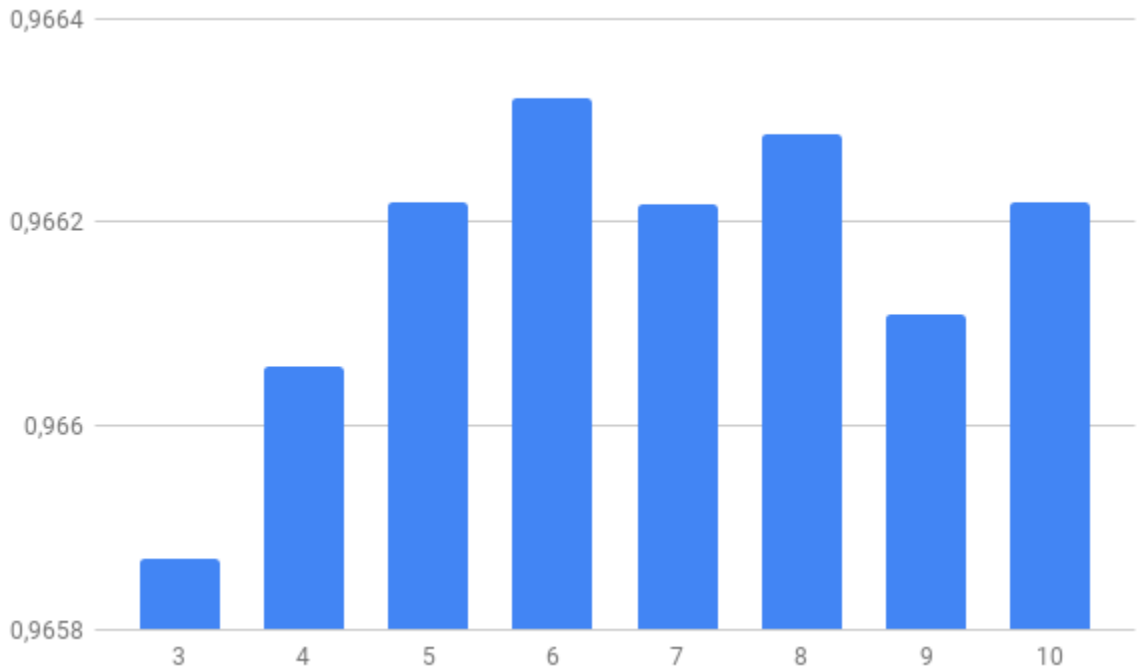
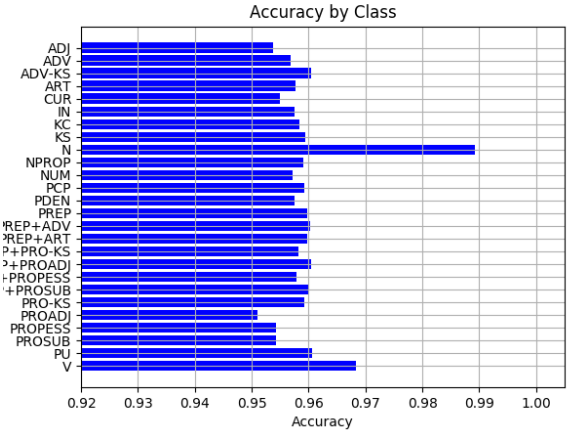


Figura 1: Comparativo da acurácia de acordo com o tamanho da janela

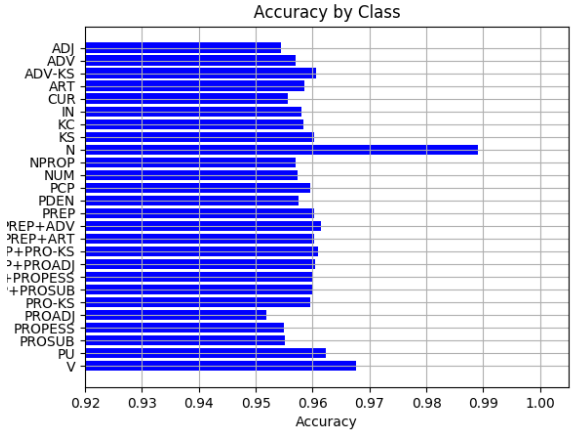
Analisando os valores, podemos ver que o tamanho de janela igual à 6 (seis) é o que apresenta a melhor acurácia, independente das classes. O fato de ter desempenho superior à a janelas maiores pode ser justificado pelo fato de que palavras mais próximas da palavra alvo são notadamente mais relacionadas à classe dela; para predizer a classe de uma palavra alvo, a classe de uma palavra que está a poucas posições dessa palavra

alvo óbviamente é um dado mais confiável do que a classe de uma palavra que está a muitas posições da palavra alvo. Ao mesmo tempo, o fato do desempenho com janela igual à 6 (seis) ser maior do que o desempenho de janelas menores pode ser justificado pelo fato de que essas janelas possuem poucas palavras para determinar a próxima em relação à ela.

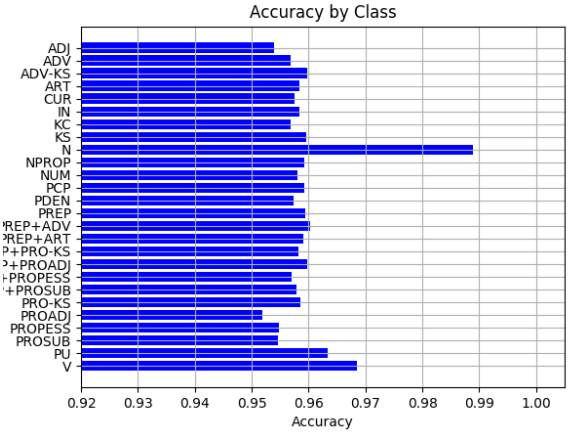
Após essa análise, o experimento seguinte consistiu em verificar a precisão da rede para cada classe, em específico. Nesse experimento os valores das janelas também foram alterados.



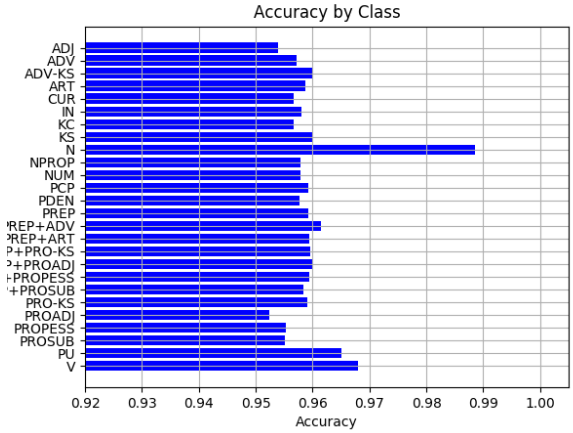
(a) Janela de tamanho igual à 3 (três)



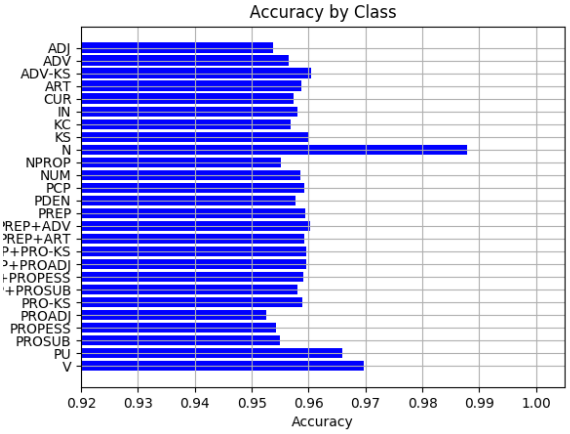
(b) Janela de tamanho igual à 4 (quatro)



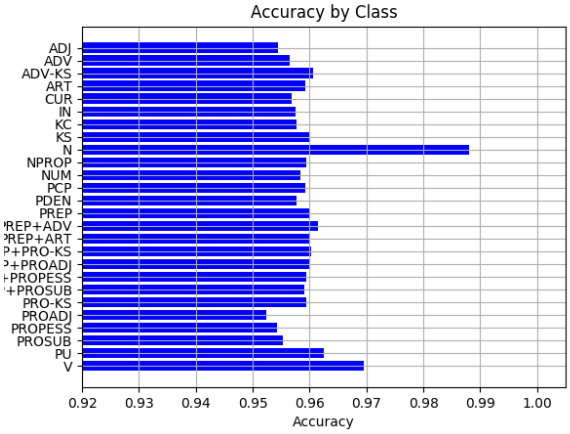
(a) Janela de tamanho igual à 5 (cinco)



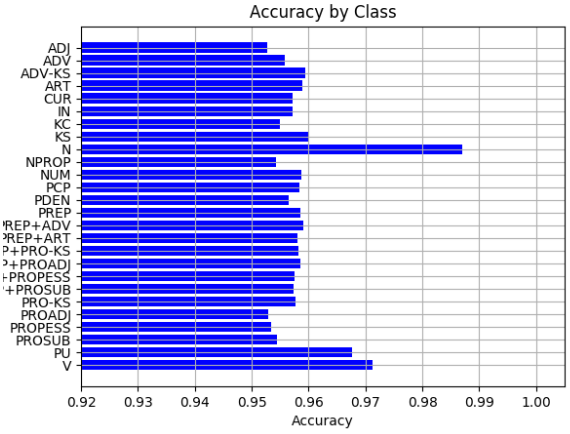
(b) Janela de tamanho igual à 6 (seis)



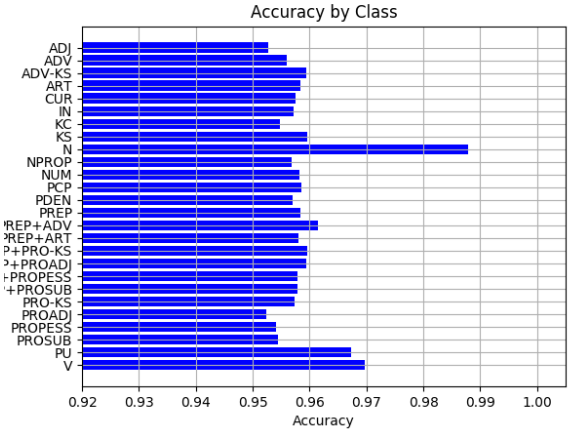
(a) Janela de tamanho igual à 7 (sete)



(b) Janela de tamanho igual à 8 (oito)



(a) Janela de tamanho igual à 9 (nove)



(b) Janela de tamanho igual à 10 (dez)

Apesar das diferentes acurácias em cada janela, que conforme visto na primeira análise a janela de tamanho 6 (seis) possui acurácia melhor, podemos ver que em cada janela os gráficos possuem estrutura semelhante: Em todos a classe gramátical *verbo* possui uma acurácia acima das demais, assim como a classe *nome*. Todas as demais classes possuem comportamento semelhante.

4 Conclusão

O trabalho analisou o uso de uma rede neural recorrente (a LSTM) para a tarefa de POS tagging, em específico aspectos como o tamanho da janela e a classe gramatical da palavra alvo.

Pelos resultados podemos chegar à conclusão de que um tamanho de janela demasiado grande é prejudicial para a predição, por conta de palavras muito distantes da palavra alvo acabarem "desvirtuando"a predição da classe da palavra alvo, e ao mesmo tempo um tamanho de janela muito pequeno também é prejudicial para predição, por conta da ausência de dados para auxiliar a rede. Também é possível notar que classes como *nome* e *verbo* são mais fácilmente identificadas dentro de uma frase, enquanto as demais classes gramaticais se mostraram parecidas em termos de facilidade de ser preditas (com uma pequena ressalva para a classe gramatical de *pronome adjetivo* que se sai um pouco pior).

Além dessas especificidades, é possível pontuar que o desempenho desse tipo de abordagem para tarefa de POS tagging possui desempenho considerável, tendo acurácia acima dos 95% independente do tamanho da janela e da classe gramatical da palavra alvo.