Processamento de Linguagem Natural

Trabalho Prático 2 - POS Tagging

Breno de Sousa Matos

Objetivo

Este trabalho tem como objetivo a implementação e avaliação de um modelo de *POS Tagging* para língua portuguesa, utilizando uma rede neural do tipo LSTM.

Introdução

Uma tarefa de *POS Tagging*, ou *Part-of-speech Tagging*, consiste em classificar palavras em relação a sua classe gramatical, como verbo, adjetivo, preposição, entre outras.

Um texto pode ser interpretado como um tipo de dado temporal, em que, para ser entendido por completo, deve-se considerar na leitura ou análise, palavras anteriores para entender o contexto atual completamente. Logo, há relação temporal entre as palavras, e, portanto, a estrutura gramatical de uma frase ou texto pode ser interpretada também de forma temporal. Logo, para a tarefa de *POS Tagging*, podemos utilizar uma abordagem usando uma LSTM.

LSTM, ou *Long Short-Term Memory*, é um tipo de rede neural recorrente (RNN) publicada por Sepp Hochreiter & Jürgen Schmidhuber, em 1997 [3]. As LSTM conseguem evitar o problema da dissipação do gradiente, e por isso são boas para desempenhar tarefas de aprendizado que necessitam de memórias de eventos que ocorreram várias iterações atrás, algo essencial para o aprendizado sobre dados temporais como texto.

Coleta dos Dados

Os dados de treino e teste foram obtidos em [1] em que cada palavra do corpus é estruturada da seguinte forma, com sua respectiva classe: “palavra”_”classe”. Complementarmente, existem 26 classes possíveis, com documentação disponível em [2].

Metodologia

Para realizar a tarefa proposta, primeiramente os dados da base de treino, com formato descrito acima, foram separados em dois corpus diferentes, um apenas com palavras e o outro somente com classes. O mesmo foi feito para a base de testes.

Em seguida, foram catalogadas as 26 possíveis classes de palavras contidas na base de treino e teste, para então representar cada uma utilizando *one-hot encoding*.

Após obter a representação one-hot de cada classe, o corpus de classes foi convertido em uma lista de vetores one-hot, para que pudesse ser interpretado e aprendido pela rede.

Utilizando o corpus de classes já convertido, foram criados dois conjuntos de dados, utilizando uma janela deslizante da seguinte forma: suponha uma janela deslizante de tamanho **n**, e o corpus de classes estruturado (como exemplo) da seguinte forma:

V ADV ADJ NUM N ART VAUX PROSUB PROPESS …

Com a janela deslizante descrita, para cada **n** palavras armazenadas em um conjunto, deveremos armazenar a palavra **n+1** no outro conjunto, a fim de obter dados de entrada e de resultado correto para predição. Isso se repete até o final do arquivo de classes.

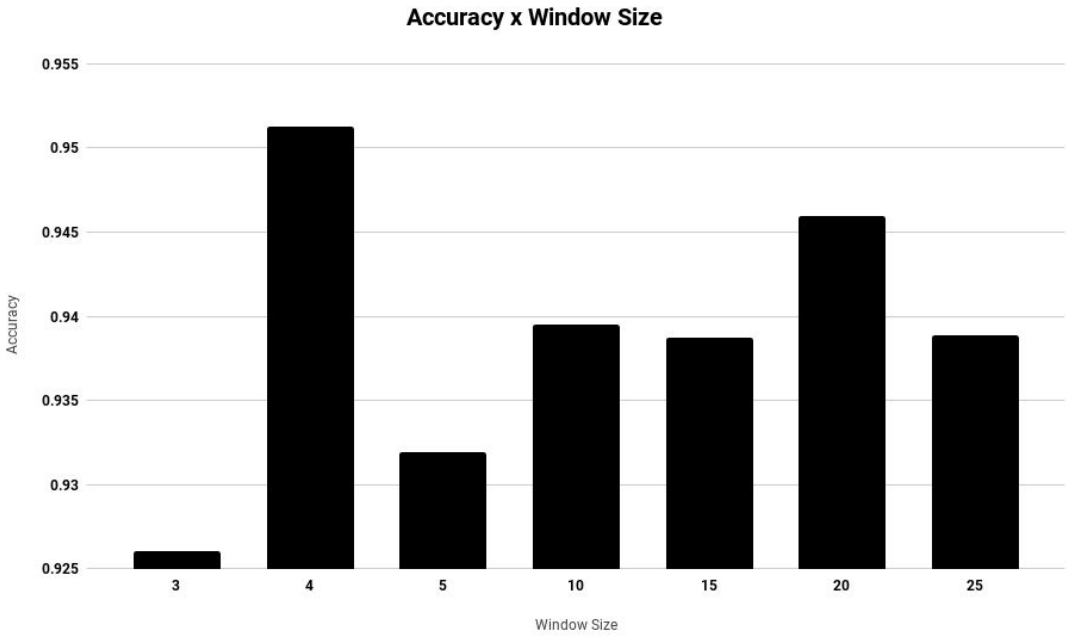
Para construir uma LSTM adequada para a tarefa proposta, foi utilizada a biblioteca Keras para Python 3. A LSTM implementada tem múltiplas camadas com funções de ativação *relu* e *sigmoid*.

Após treinado o modelo, foi utilizado o corpus de teste para avaliá-lo. O objetivo era avaliar não só a acurácia do modelo de forma geral, contabilizado acertos totais, como também em relação a cada uma das 26 classes gramaticais. Para avaliar a acurácia total, foi utilizada uma janela deslizante de forma idêntica à descrita anteriormente. Para avaliar a acurácia por classe, foi construído um dicionário com as 26 classes como chaves. Neste dicionário, cada chave tinha como correspondente uma lista de várias janelas deslizantes. Ou seja, foram armazenados todos os conjuntos de janelas para as quais a próxima classe era a sua respectiva chave no dicionário, para todas as classes.

Análise Experimental

Para avaliação do modelo, foram utilizadas janelas deslizantes de tamanhos 3,4,5,10,15,20 e 25.

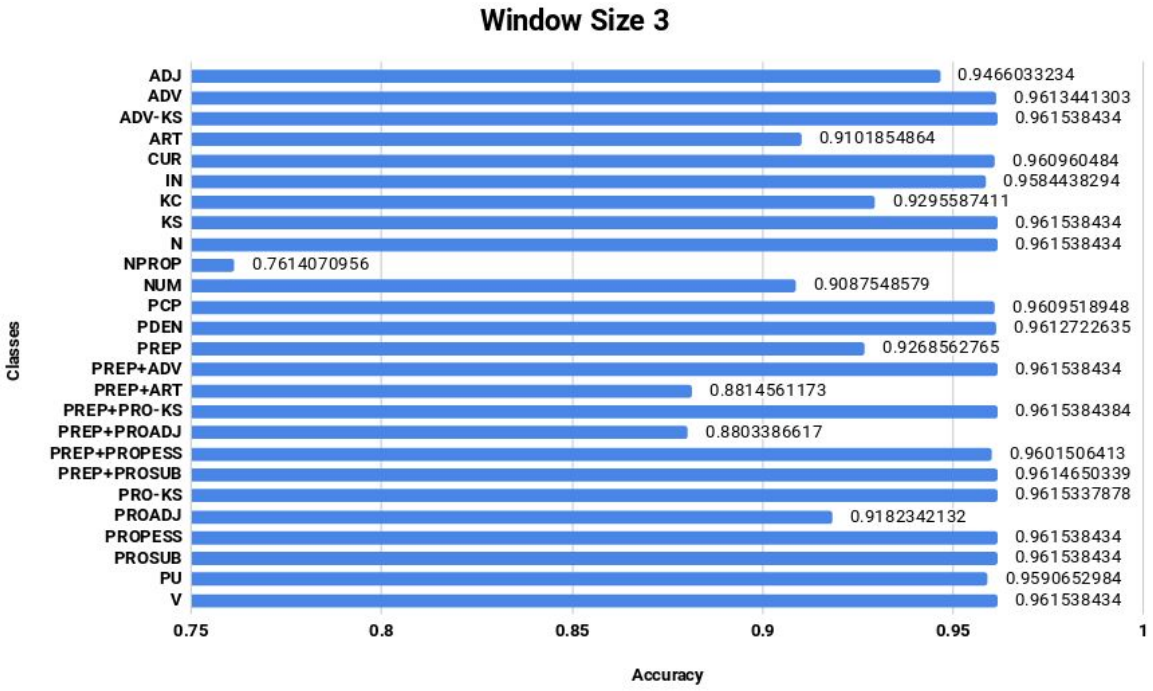
Primeiramente, temos os resultados da acurácia total do modelo, por tamanho de janela deslizante, no gráfico abaixo:

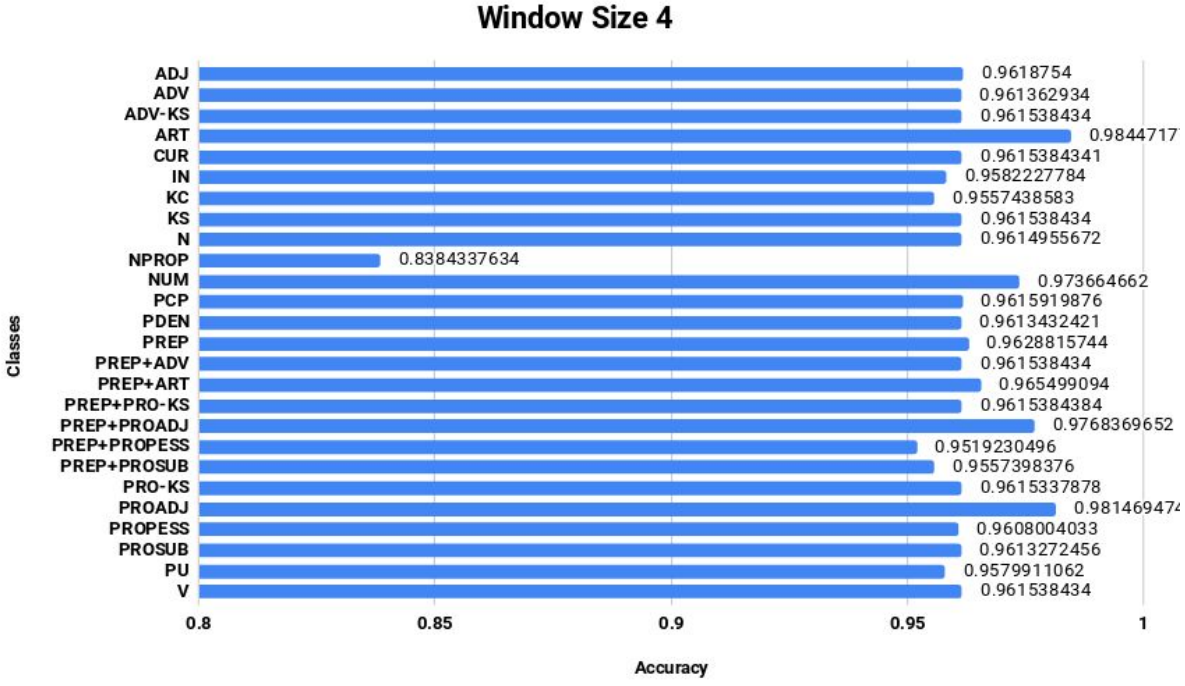


Como é possível observar, o modelo com janela deslizante de tamanho 4 obteve o melhor resultado entre os 7 tamanhos de janela utilizados nos experimentos. Este resultado pode ser interpretado da seguinte forma: as palavras mais relevantes para predizer à qual classe gramatical uma dada palavra P pertence são as mais próximas desta, em contraste à palavras muito distantes, que, apesar de importante para contextualização e entendimento

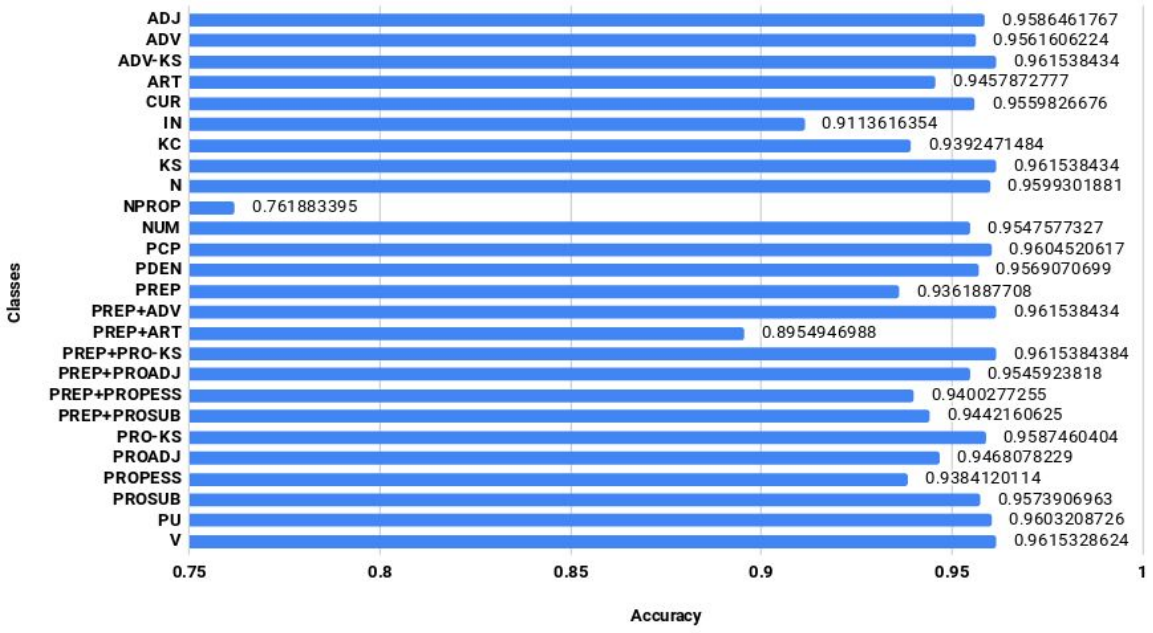
do texto, podem não ser tão relevantes para determinar qual classe gramatical a palavra alvo da predição pertence.

Em seguida, foram gerados gráficos referentes à acurácia para cada classe gramatical, por tamanho de janela deslizante, mostrados abaixo:

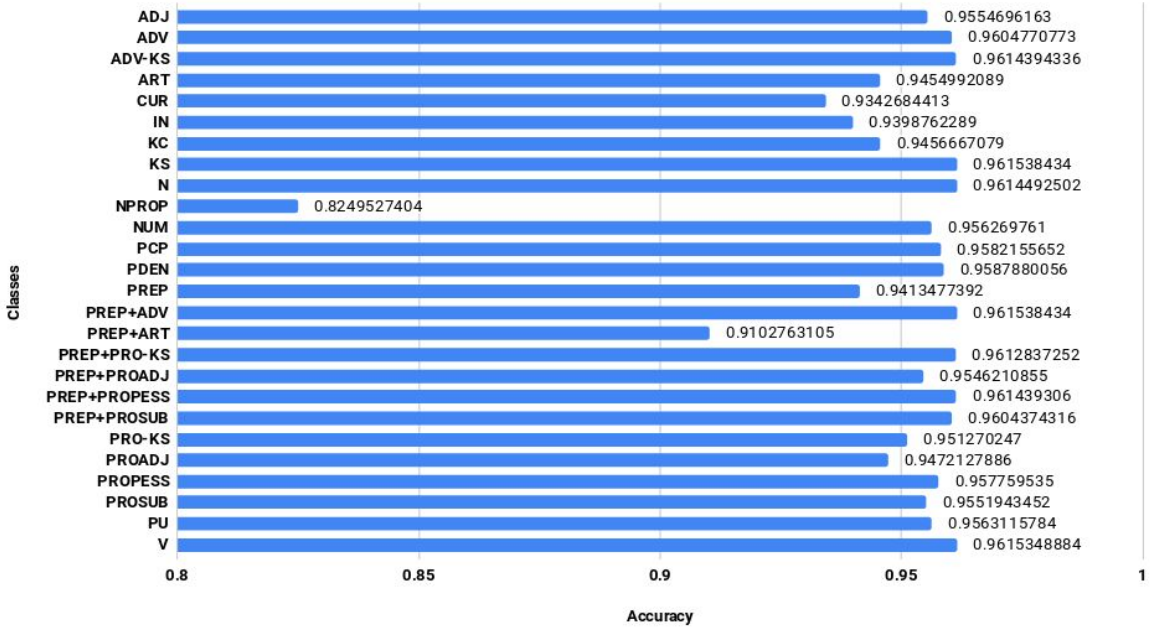


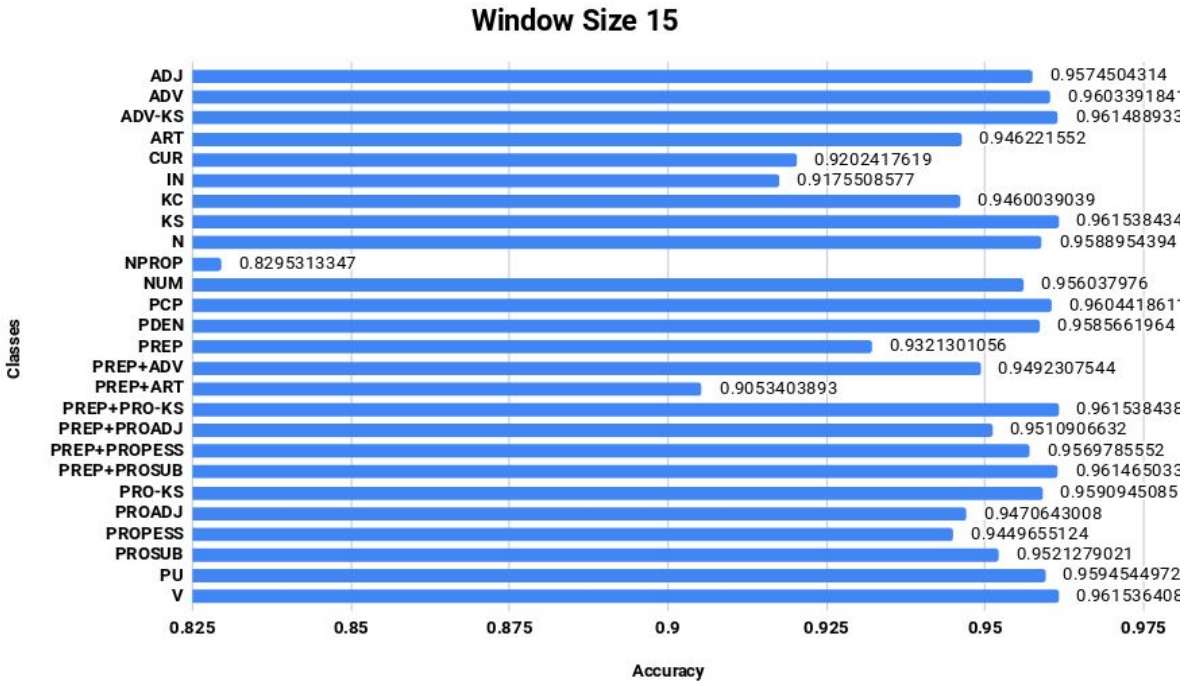


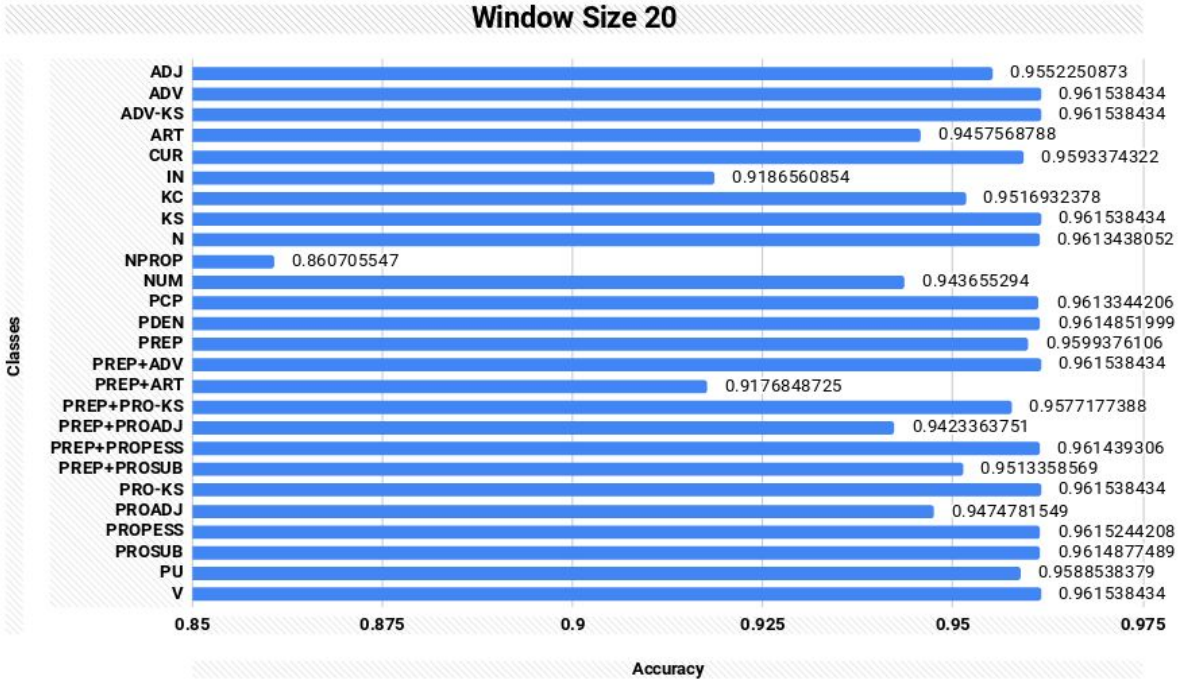
Window Size 5

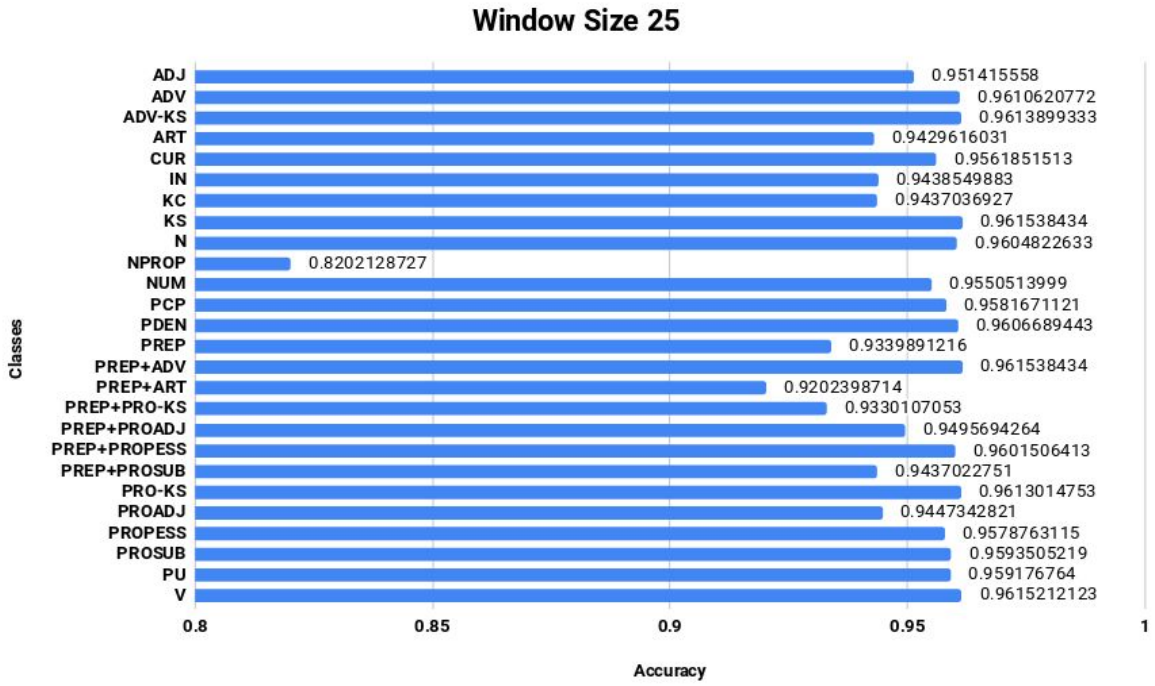


Window Size 10



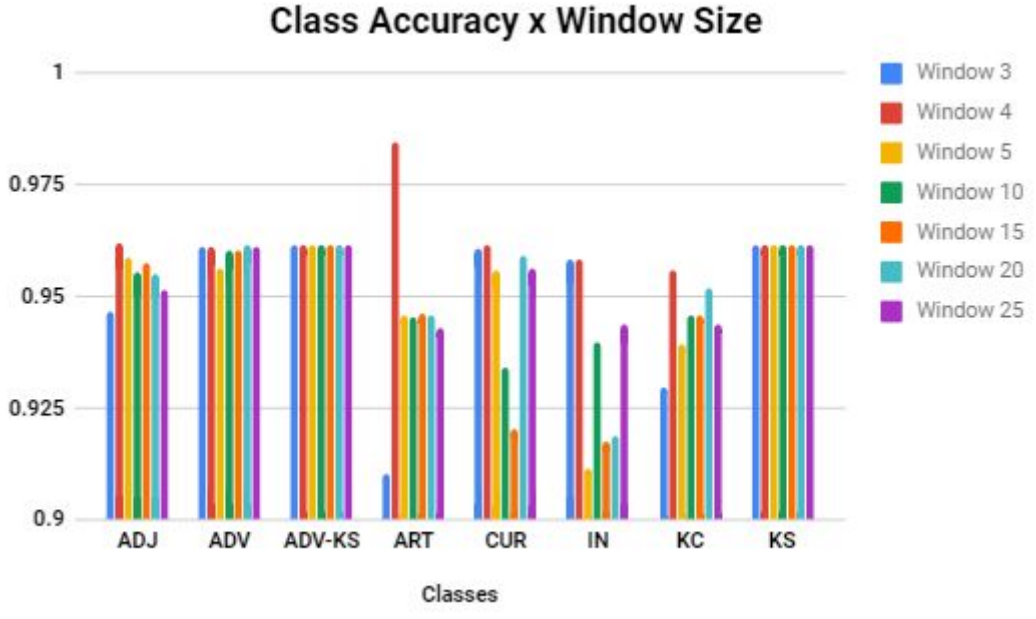

Window Size 15
ADJ 0.9574504314
ADV 0.960339184
ADV-KS 0.961488933
ART 0.946221552
CUR 0.9202417619
IN 0.9175508577
KC 0.9460039039
KS 0.961538434
N 0.9588954394
NPROP 0.8295313347
NUM 0.956037976
PCP 0.960441861
PDEN 0.9585661964
PREP 0.9321301056
PREP+ADV 0.9492307544
PREP+ART 0.9053403893
PREP+PRO-KS 0.961538438
PREP+PROADJ 0.9510906632
PREP+PROPESS 0.9569785552
PREP+PROSUB 0.961465033
PRO-KS 0.9590945085
PROADJ 0.9470643008
PROPESS 0.9449655124
PROSUB 0.9521279021
PU 0.9594544972
V 0.961536408
Classes
0.825
0.85
0.875
0.9
0.925
0.95
0.975
Accuracy

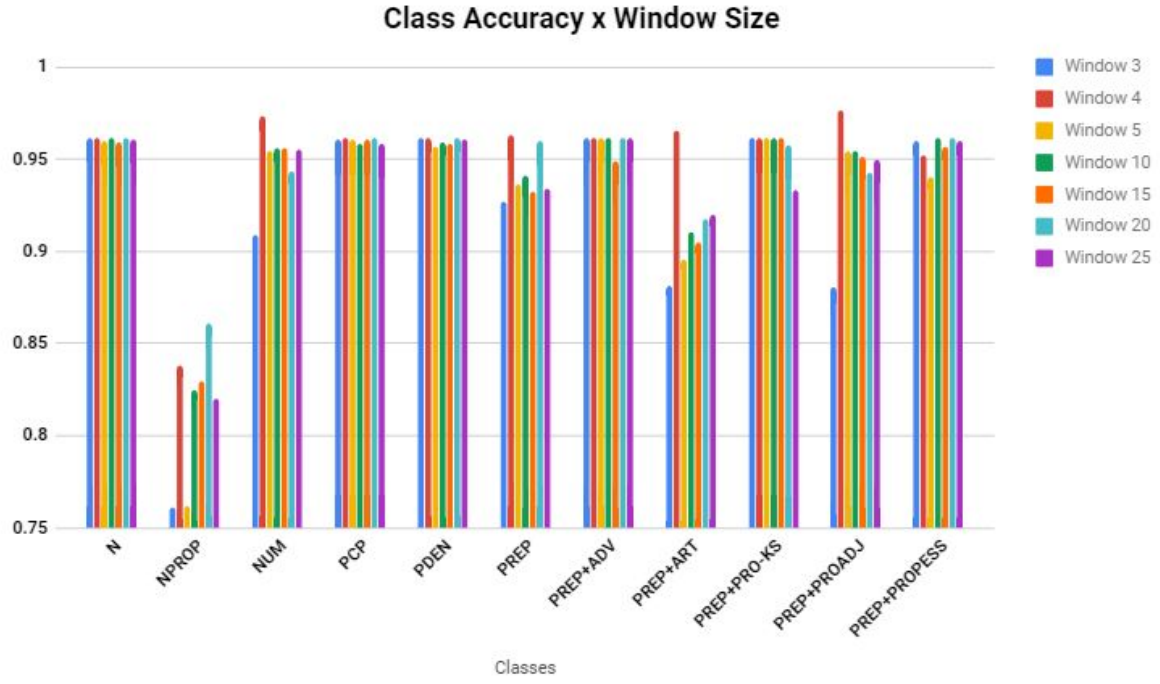

Window Size 20
ADJ 0.9552250873
ADV 0.961538434
ADV-KS 0.961538434
ART 0.9457568788
CUR 0.9593374322
IN 0.9186560854
KC 0.9516932378
KS 0.961538434
N 0.9613438052
NPROP 0.860705547
NUM 0.943655294
PCP 0.9613344206
PDEN 0.9614851999
PREP 0.9599376106
PREP+ADV 0.961538434
PREP+ART 0.9176848725
PREP+PRO-KS 0.9577177388
PREP+PROADJ 0.9423363751
PREP+PROPESS 0.961439306
PREP+PROSUB 0.9513358569
PRO-KS 0.961538434
PROADJ 0.9474781549
PROPESS 0.9615244208
PROSUB 0.9614877489
PU 0.9588538379
V 0.961538434
Classes
0.85
0.875
0.9
0.925
0.95
0.975
Accuracy



Os gráficos acima mostram que, consistentemente, a classe de nomes próprios (**NPROP**) tem os piores resultados de predição, obtendo os piores resultados absolutos para todos os tamanhos de janela. Este resultado pode ser entendido da seguinte forma: nomes próprios podem ocorrer com menos frequência em relação às demais classes, assim como criar ambiguidades, como o nome Santos, que pode ser tanto nome próprio quanto adjetivo.

Resultados, para todas as classes, para todos os tamanhos de janela utilizados, podem ser visualizados nos gráficos abaixo:





Nos gráficos acima, é possível observar que algumas classes, como verbo (**V)** e advérbio (**ADV**) têm suas respectivas acurácias de previsão quase inalteradas mesmo variando o tamanho da janela. Isso pode ser entendido com a alta frequência de palavras dessas classes na língua portuguesa, assim como serem peças centrais para construção de frases, visto que se relacionam sintaticamente com diversas outras classes.

Por outro lado, classes como artigo (**ART**), numeral (**NUM**), preposição + artigo (**PREP+ART**) e preposição + pronome adjetivo (**PREP+PROADJ**) beneficiam-se de janelas de tamanhos menores, obtendo seus melhores resultados de predição com a janela de

tamanho 4. Este resultado pode ser explicado pois essas classes dependem intrinsecamente das classes das palavras próximas à elas, visto que um artigo, por exemplo, faz referência à uma palavra que vem logo em seguida, ou então poucas palavras adiante, não se beneficiando portanto de uma janela de tamanho 20 ou 25, por exemplo.

Conclusão

Este trabalho teve como objetivo implementar e avaliar um modelo de POS Tagging utilizando uma LSTM. Para tal, foram utilizados diversos tamanhos de janela deslizante. Os resultados obtidos mostram que janelas de tamanhos muito grandes têm resultados piores que janelas de tamanhos menores, no geral. Como discutido acima, o melhor resultado geral foi obtido utilizando uma janela de tamanho 4, justificado pela relação intrínseca entre palavras próximas para o entendimento e predição de classes de palavras. Além disso, para algumas classes (como verbo), o tamanho da janela não é tão decisivo para a acurácia final, que mantêm-se quase inalterado mesmo para janelas de tamanhos diferentes.

Por fim, é possível observar que a tarefa proposta foi realizada com taxas de acurácia altas, com todos os modelos produzindo valores acima de 90% para acurácia geral.

Referências

[1] http://nilc.icmc.usp.br/macmorpho/macmorpho-v3.tgz
[2] http://nilc.icmc.usp.br/macmorpho/macmorpho-manual.pdf
[3] https://dl.acm.org/citation.cfm?id=1246450