Universidade Federal do Rio Grande do Norte

Instituto Metrópole Digital

DIM0135 - Processamento de Linguagem Natural

# Part-of-Speech Tagger: Unigramas e Bigramas

João Vitor Venceslau Coelho

Natal/RN

2021

## Introdução

Este relatório tem como objetivo apresentar as particularidades das duas abordagens selecionadas para a tarefa de **Part-of-Speech Tagging**, uma utilizando **unigramas** e a outra **bigramas**. Ao fim são apresentados os resultados da aplicação das duas abordagens em um corpus selecionado (Penn Treebank) e uma breve discussão sobre os resultados obtidos nas duas estratégias.

## Estratégias Usadas

### Unigramas

Durante o treinamento, o primeiro tratamento após obter o par token-tag é transformar o token para sua versão com apenas letras minúsculas, em seguida, ocorre o processo comum de contagem de ocorrências de par token-tag.

Após o treino é computada a tag que deve ser utilizada quando um token desconhecido for visto durante o teste, para esta situação os tokens que aparecem menos de *10* vezes no conjunto de treino são agrupados, e a tag que mais ocorrer nesse agrupamento será a tag utilizada na classificação de tokens desconhecidos.

Durante o teste, após transformar o token para sua versão com apenas letras minúsculas é feito o teste se o token é conhecido ou não, se for conhecido, para classificar esse token é usada a tag mais vista junta a ele durante o treino, caso contrário, se o token for composto apenas de números, é utilizada a tag *‘CD’*e se também não for apenas números no token, é usada a tag computada anteriormente para tokens desconhecidos.

### Bigramas

Durante o treinamento, no início de cada sentença, são atribuídos como token e tag anteriores a string *‘PS’*, em seguida, para cada par token-tag obtido da sentença é feita a transformação do token para a versão em minúsculas. Então são computadas as devidas estatísticas para o par token-tag, considerando também o par token-tag anterior, assim obtemos quantas vezes determinado token aparece após outro token com determinada tag.

Após esse treino são computadas algumas informações para lidar com uma sequência de tokens desconhecida, por exemplo, dois tokens conhecidos mas que não foram vistos juntos durante o treino, ou um token totalmente desconhecido, nunca visto antes. Para sequências de tokens não vistas antes, utiliza-se apenas a tag mais comumente associada a esse token, sem considerar o token anterior, como se fosse um unigrama, e para utilizar em tokens totalmente novos, utiliza-se a mesma estratégia descrita para unigramas.

Durante o teste, caso a sequência de tokens seja conhecida utiliza-se a tag mais vista para a sequência durante o treino, caso seja uma sequência nova, mas com um token conhecido, utiliza-se a informação computada anteriormente, como se fosse um unigrama, e para tokens totalmente novos, assim como foi feito em unigramas, verifica se ele é composto só de números, se for, atribui a tag *‘CD’*, caso contrário utiliza-se a tag computada para tokens totalmente novos.

## Resultados Obtidos

Foi utilizado o corpus do Penn Treebank para realizar os testes, a versão disponibilizada pelo Moodle onde a extração dos pares token-tag válidos já estava realizada, mas de qualquer forma foi implementado também um pequeno script que realiza essa extração na outra versão do Penn Treebank que foi disponibilizada. As seções de 0 a 18 foram utilizadas para o treinamento, as seções 19, 20 e 21 para a validação e as 22, 23 e 24 para o teste final, assim como foi sugerido no Moodle.

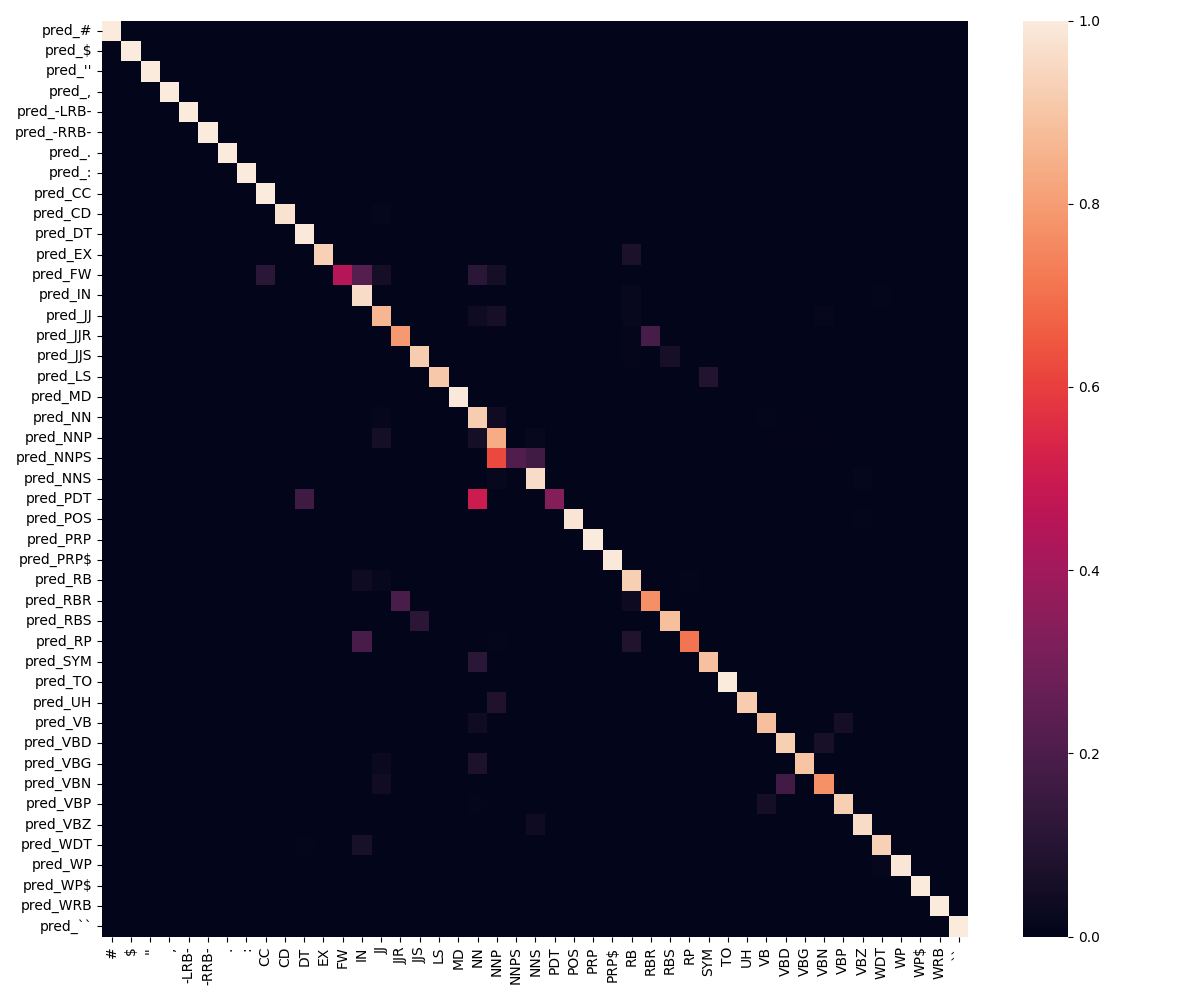
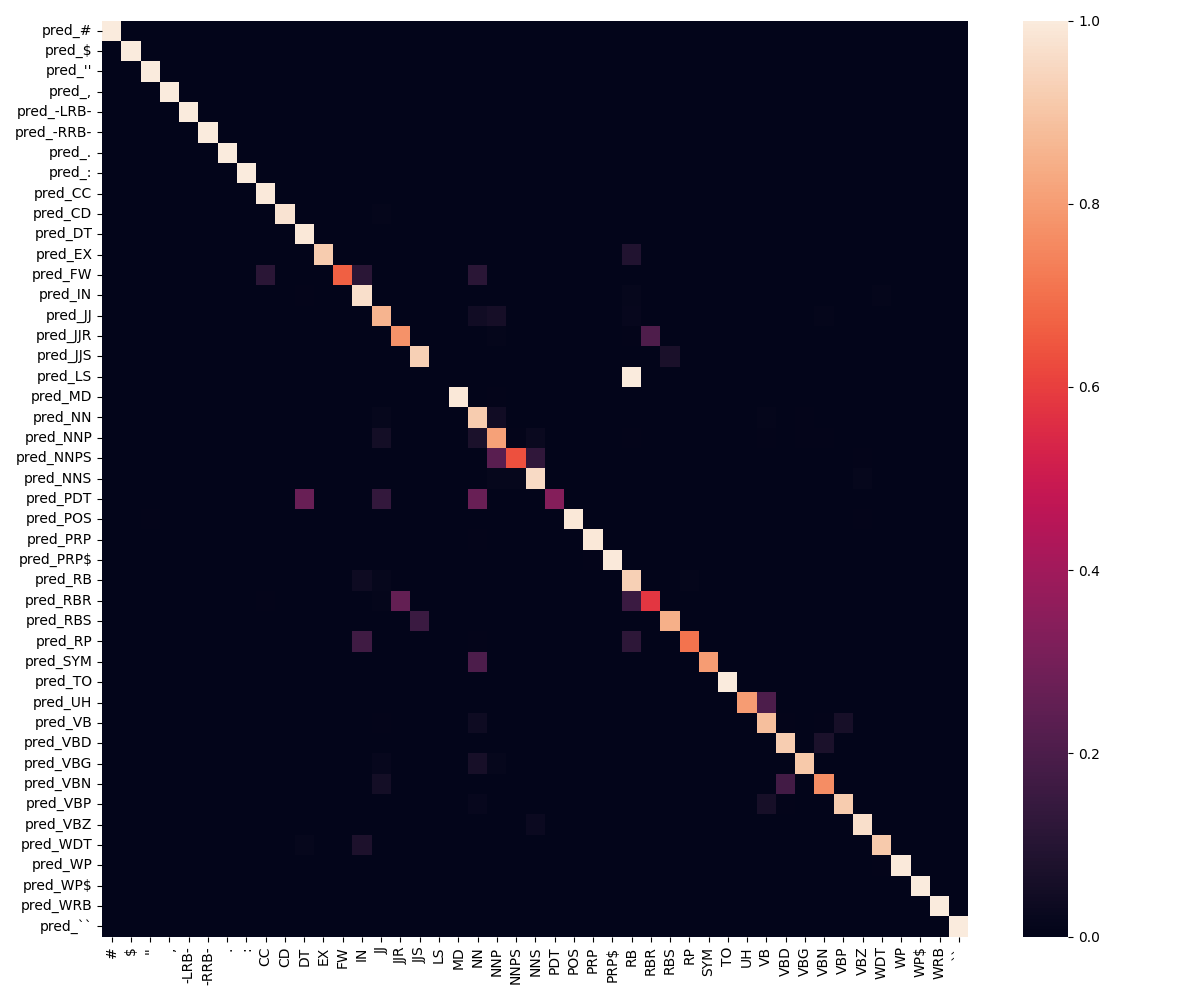
As matrizes de confusão apresentadas podem ser consultadas em detalhes com os arquivos .csv anexados ao relatório. Aqui serão mostradas apenas visualizações simplificadas.

### Matriz de Confusão - Unigrama Validação e Teste (Com lower)

### 

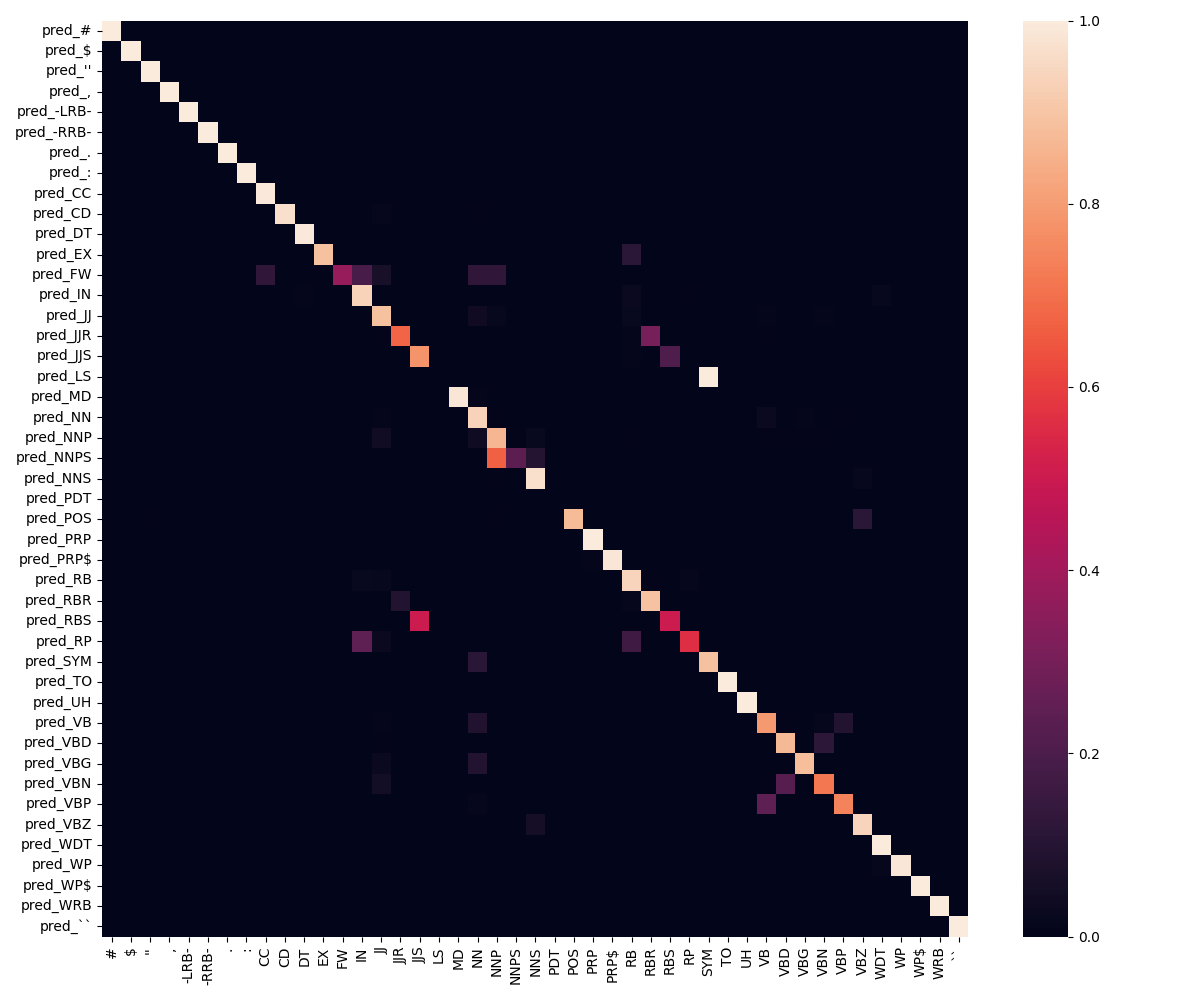
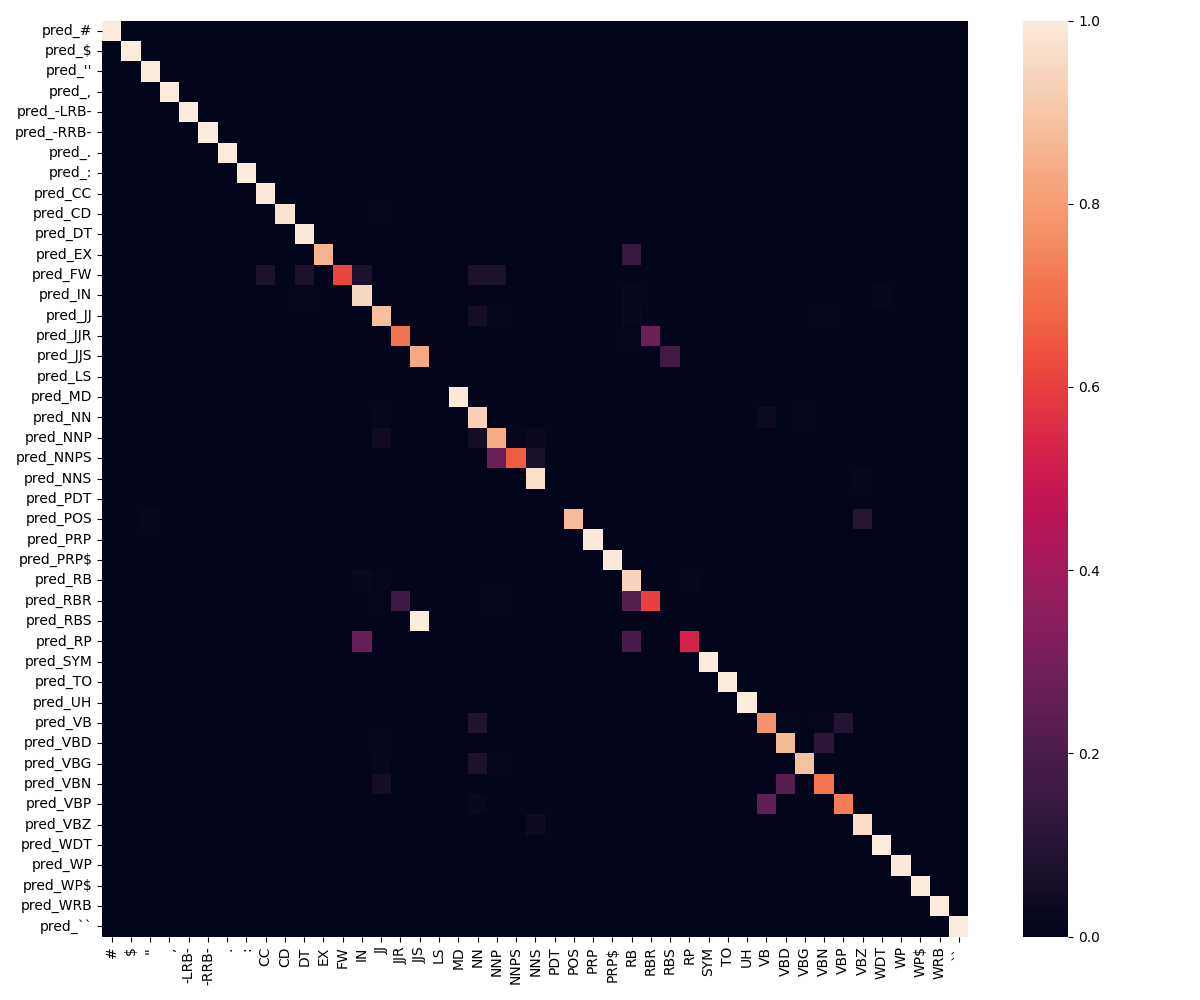
A esquerda: Matriz de confusão para a validação. A direita: Matriz de confusão para o teste.

### Matriz de Confusão - Bigrama Validação e Teste (Com lower)



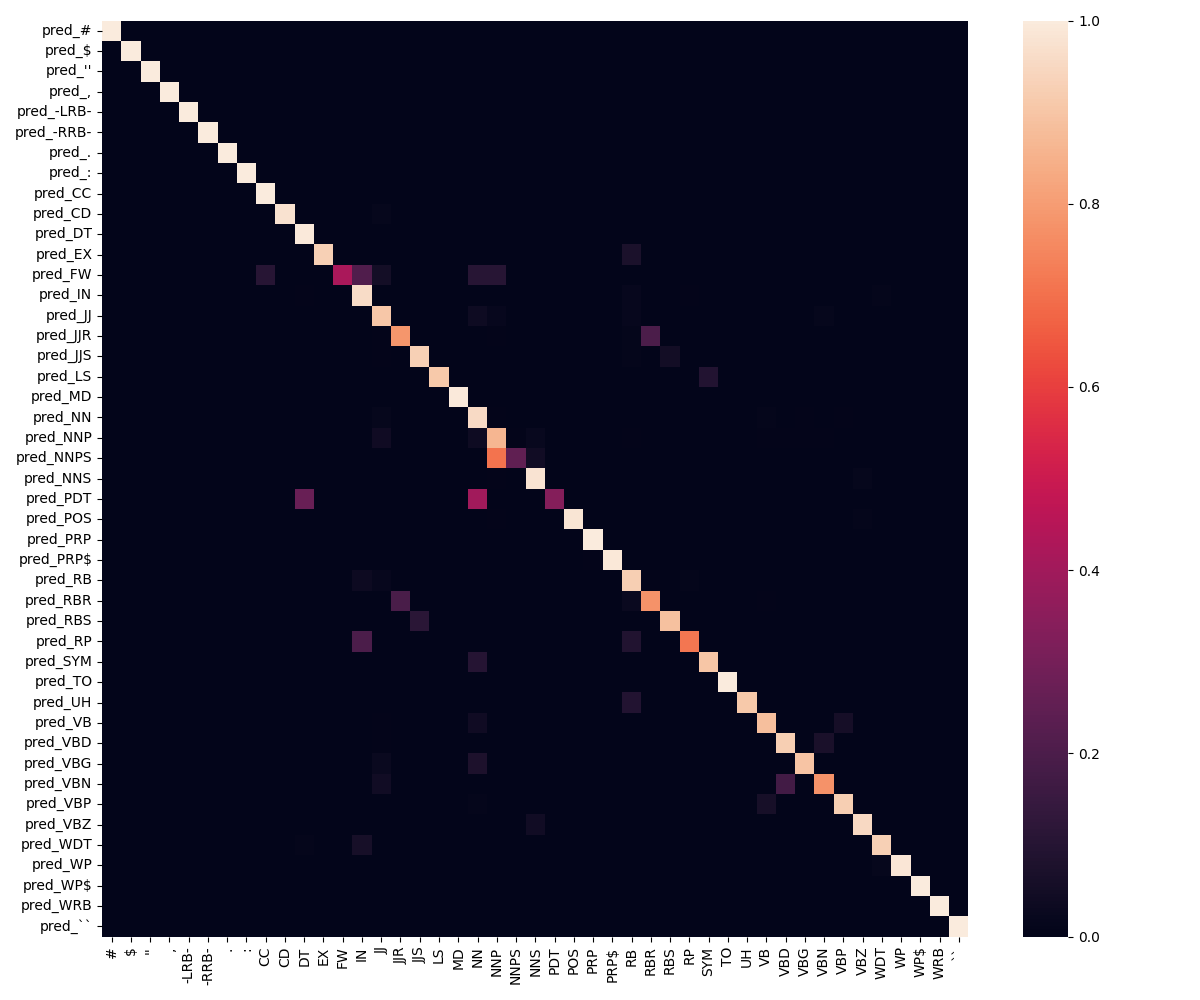
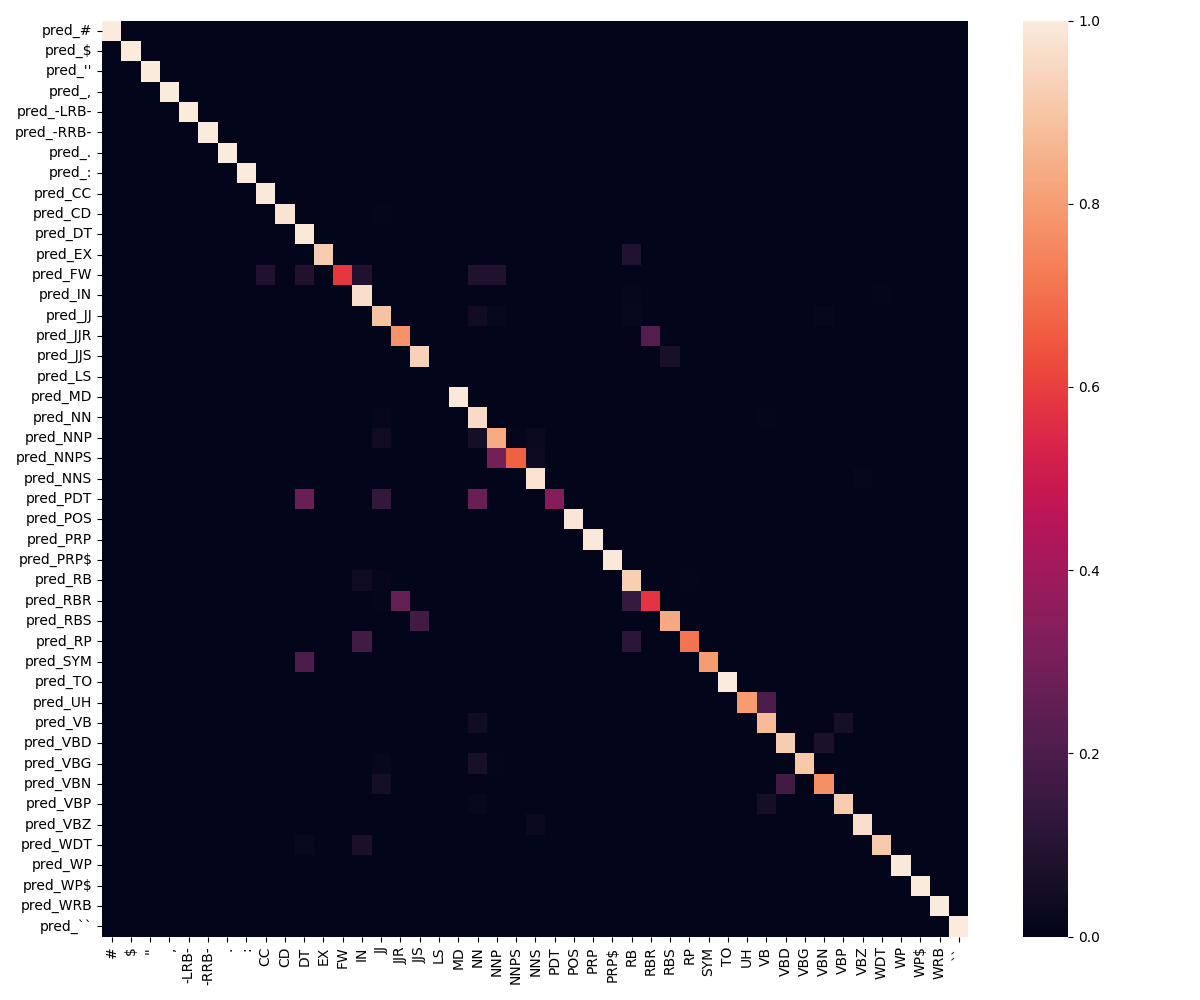
A esquerda: Matriz de confusão para a validação. A direita: Matriz de confusão para o teste.

### Matriz de Confusão - Unigrama Validação e Teste (Sem lower)



A esquerda: Matriz de confusão para a validação. A direita: Matriz de confusão para o teste.

### Matriz de Confusão - Bigrama Validação e Teste (Sem lower)



A esquerda: Matriz de confusão para a validação. A direita: Matriz de confusão para o teste.

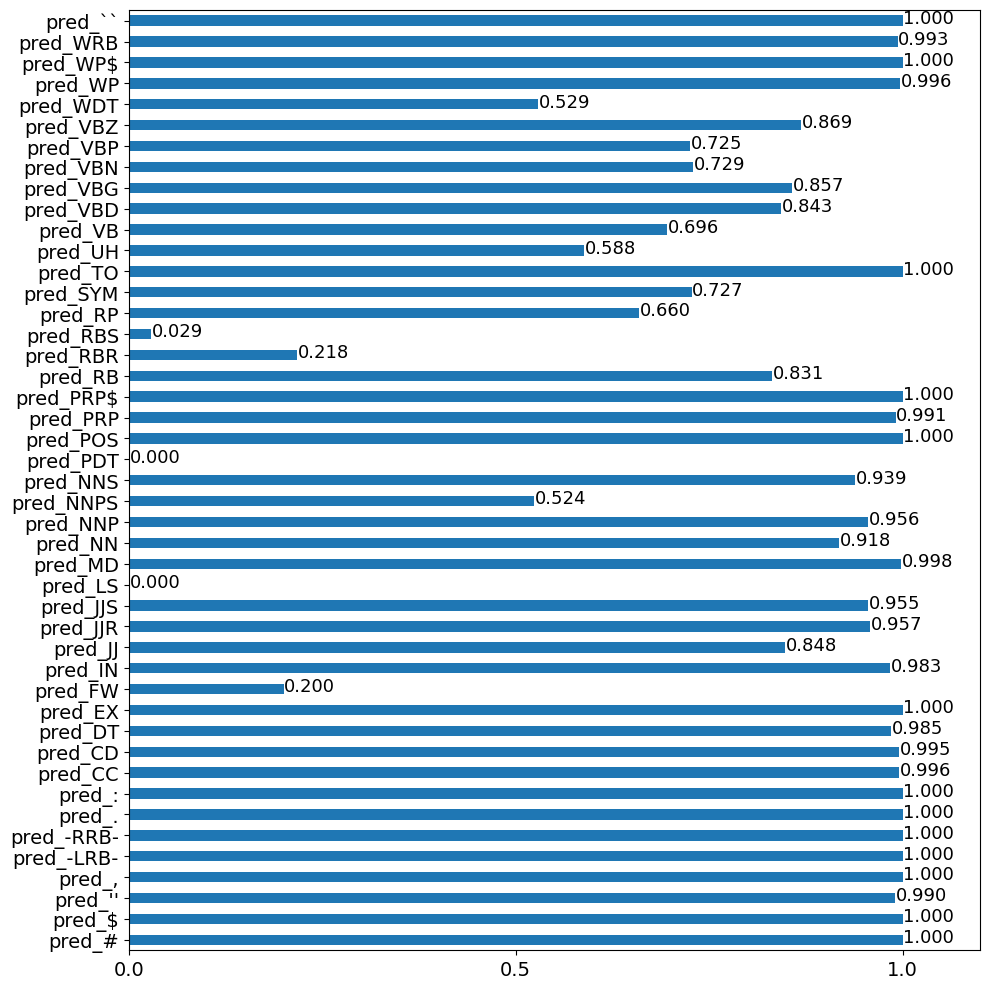
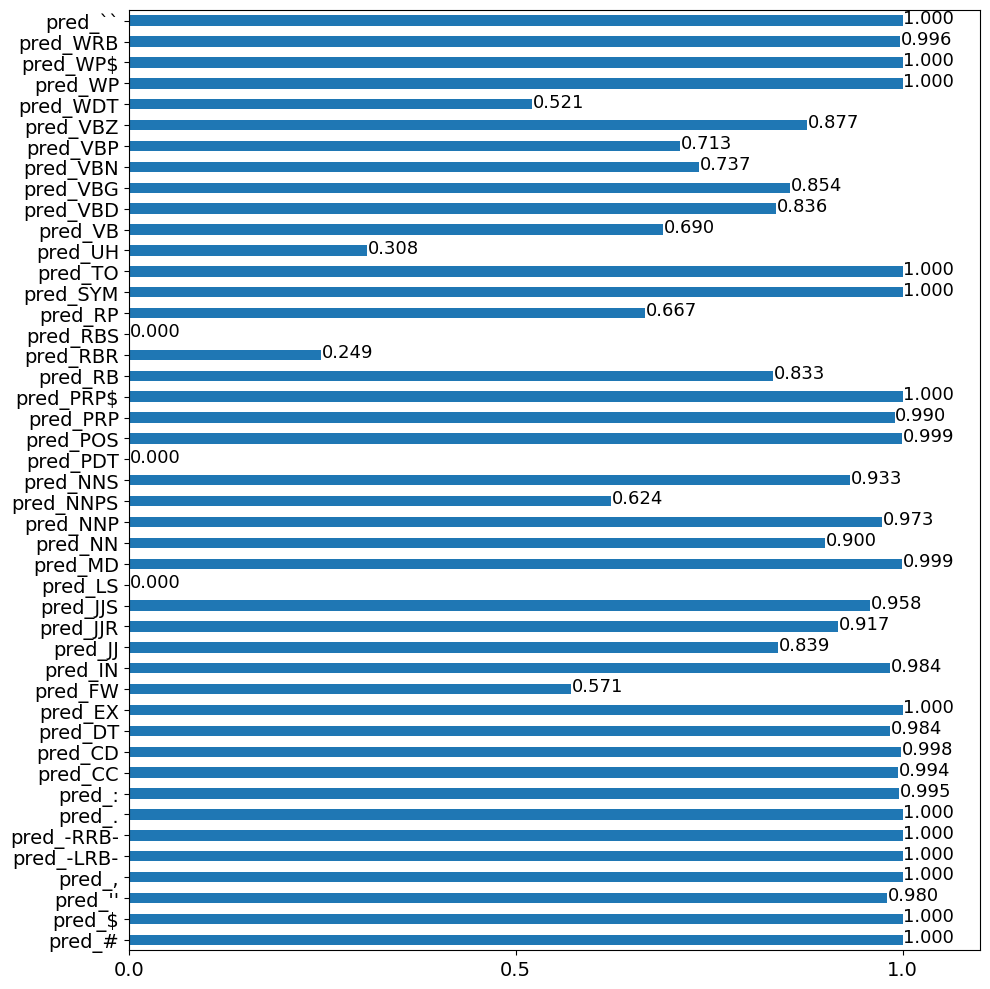
## Discussões dos Resultados

Comparando as matrizes de confusão obtidas ao utilizar unigramas com as obtidas ao utilizar bigramas, realizando o pré-processamento aplicando o método lower nos tokens, percebemos facilmente que ao usar o unigrama as tags ‘LS’ e ‘PDT’ nunca são usadas, enquanto que com o uso de bigramas essas tags passam a ser utilizadas, mesmo que não corretamente em todos os casos. Na matriz de confusão obtida para o uso de bigramas, no conjunto de validação, destaca-se o uso incorreto da tag ‘LS’ no lugar da tag ‘RB’, porém o mesmo não ocorreu no conjunto de testes, a tag passou a ser classificada de forma muito mais coerente com o esperado. Em compensação, outras tags como ‘NNPS’ tiveram um aumento nas classificações incorretas, principalmente sendo classificadas como ‘NNP’. Algo similar ocorreu com as tags ‘PDT’ e ‘NN’, onde ‘PDT’ passou a ser incorretamente classificada como ‘NN’ mais vezes no conjunto de teste quando comparado com o conjunto de validação.

Enquanto que nas matrizes de confusão obtidas com unigramas não realizando o pré-processamento dos tokens com o lower, no conjunto de validação, os erros para a tag ‘RBS’ aumenta consideravelmente, enquanto que no conjunto de teste, a tag ‘LS’ que antes nem sequer era usada com unigramas realizando o pré-processamento com o lower, passa a ser utilizada, mesmo que incorretamente.

Para as matrizes de confusão obtidas com bigramas, na validação a tag ‘LS’ não é utilizada, enquanto que no teste ela é usada corretamente, outras alterações também ocorrem, porém essa é a mais perceptível. Por fim, para efeitos de comparação foram calculadas algumas métricas, como *precision* e *recall* em cada um dos cenários:

### Precision - Unigrama Validação e Teste (Sem lower)



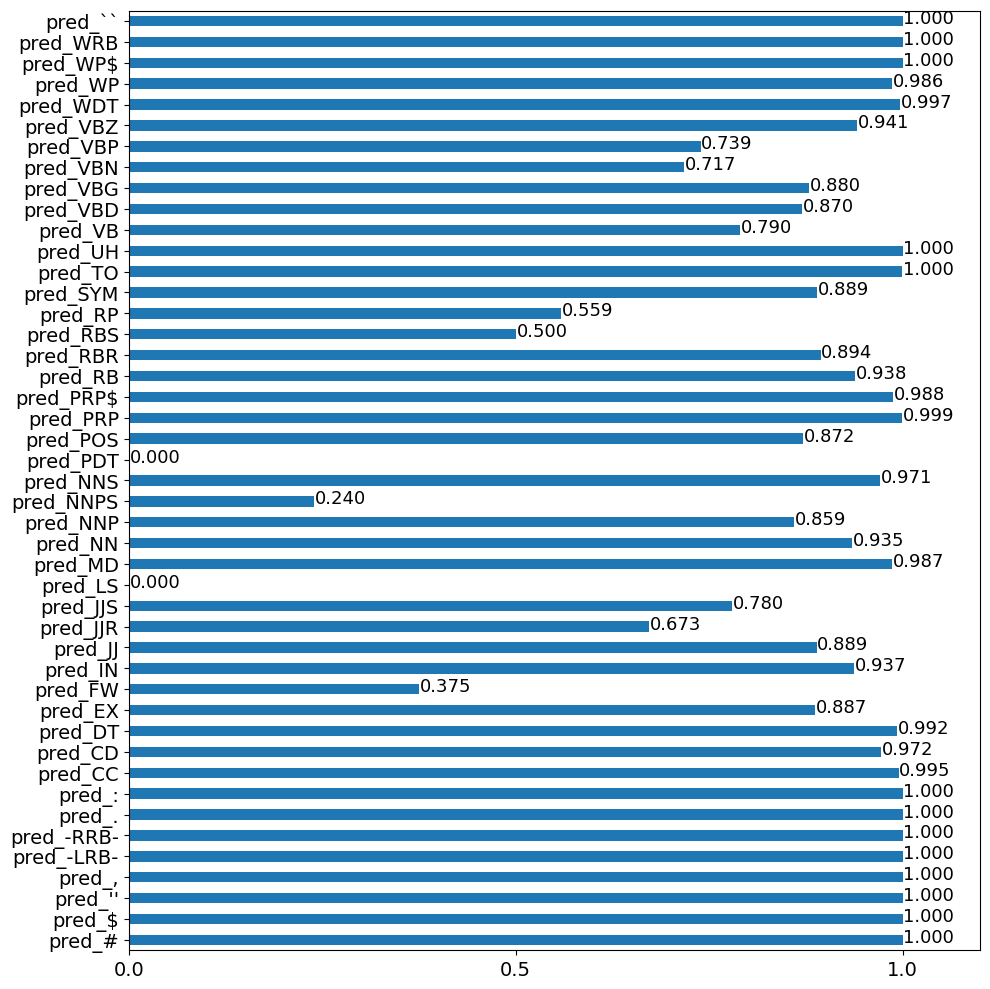
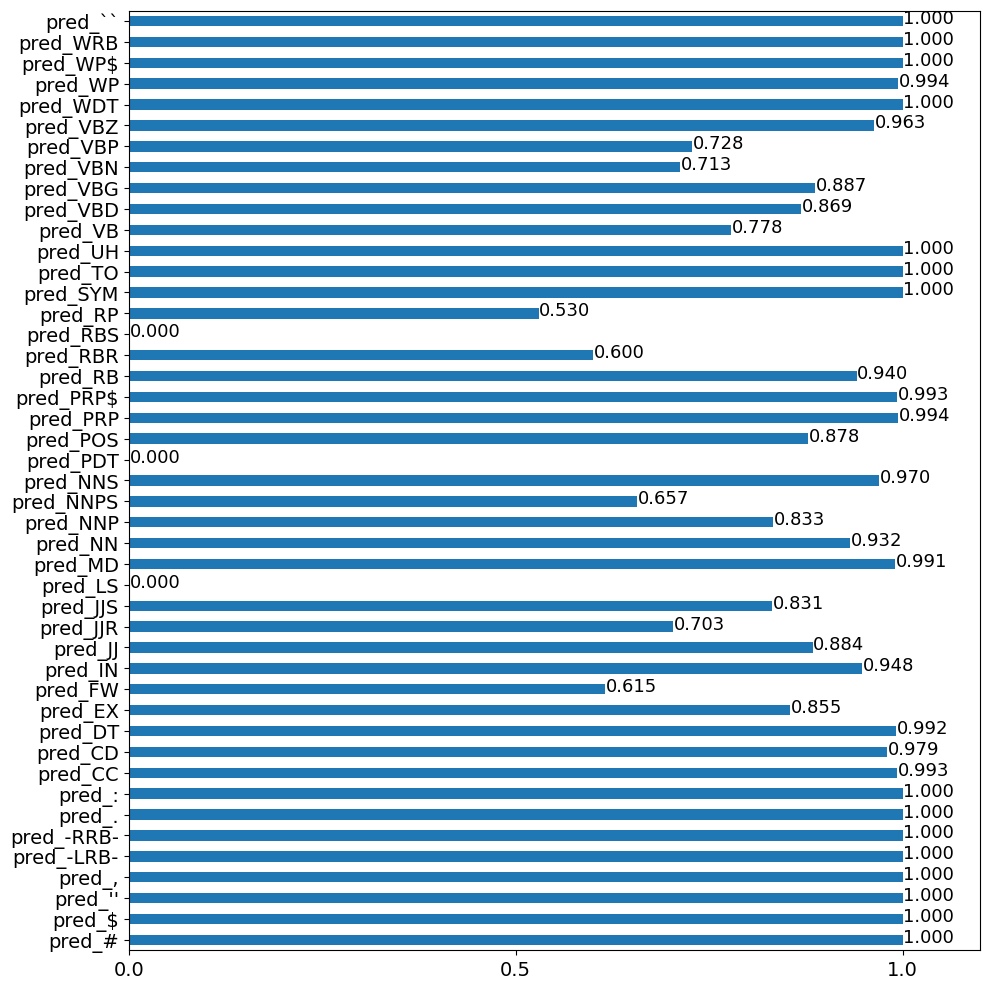
A esquerda: Precision na validação. A direita: Precision no teste.

### Precision - Bigrama Validação e Teste (Sem lower)

### 

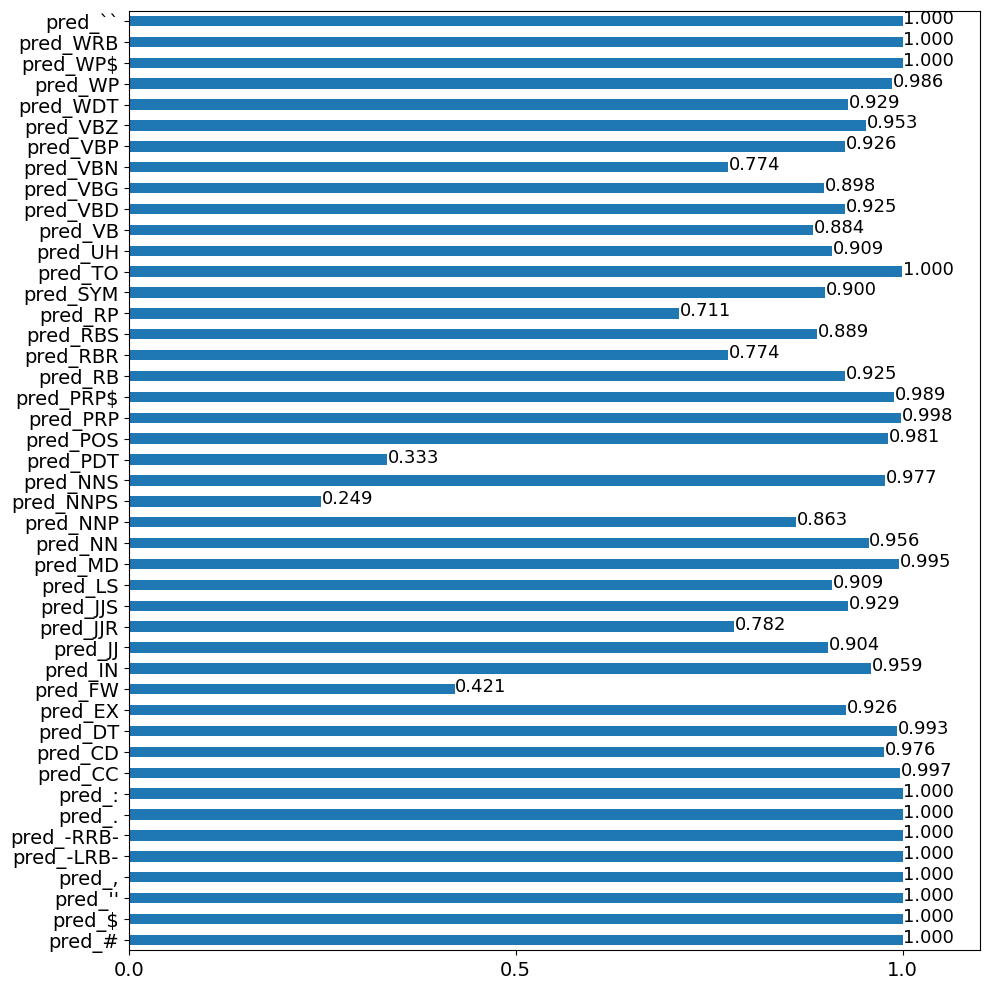
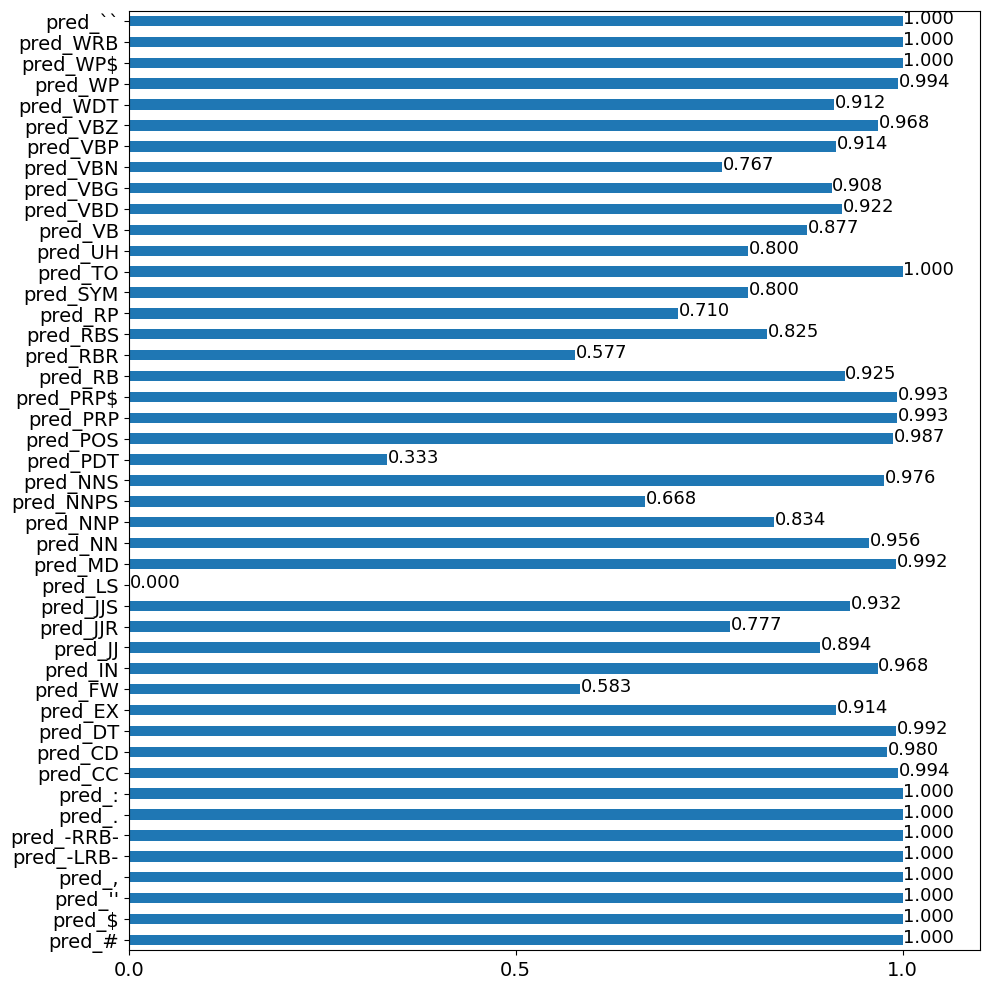
A esquerda: Precision na validação. A direita: Precision no teste.

### Recall - Unigrama Validação e Teste (Sem lower)



A esquerda: Recall na validação. A direita: Recall no teste.

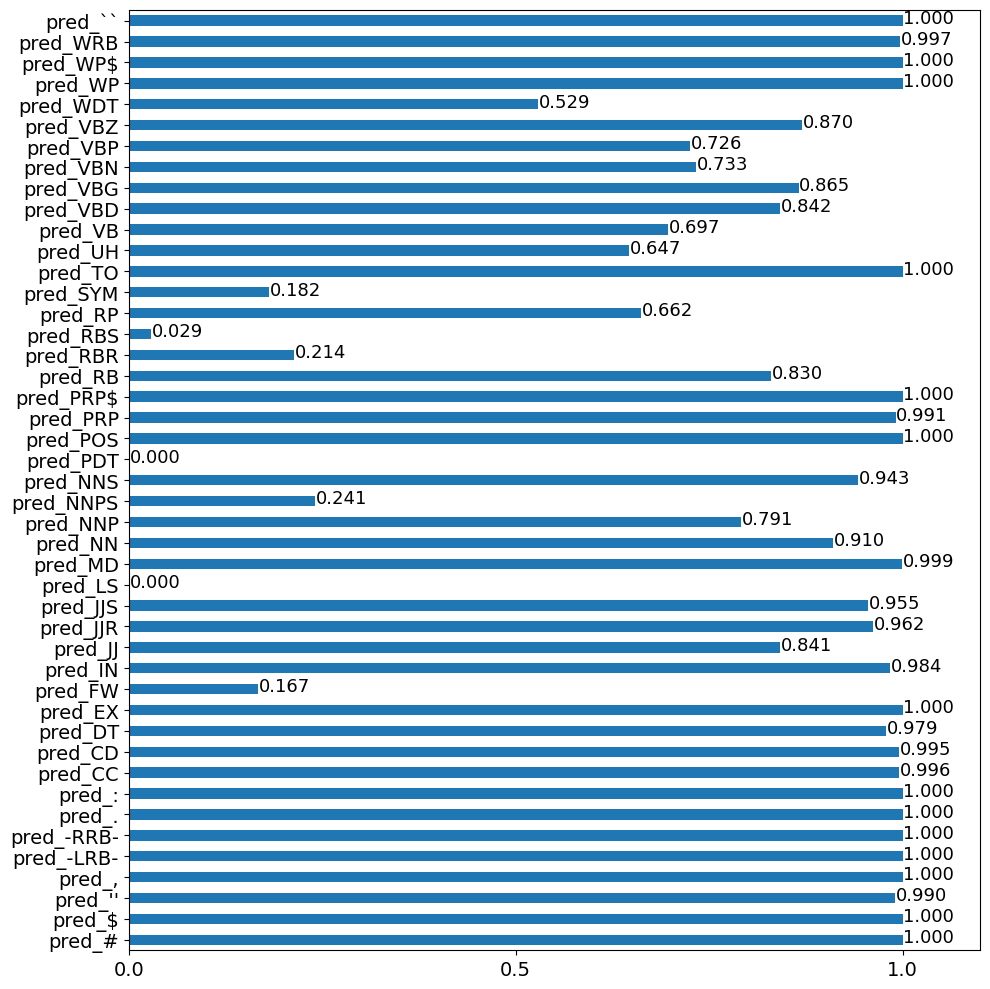
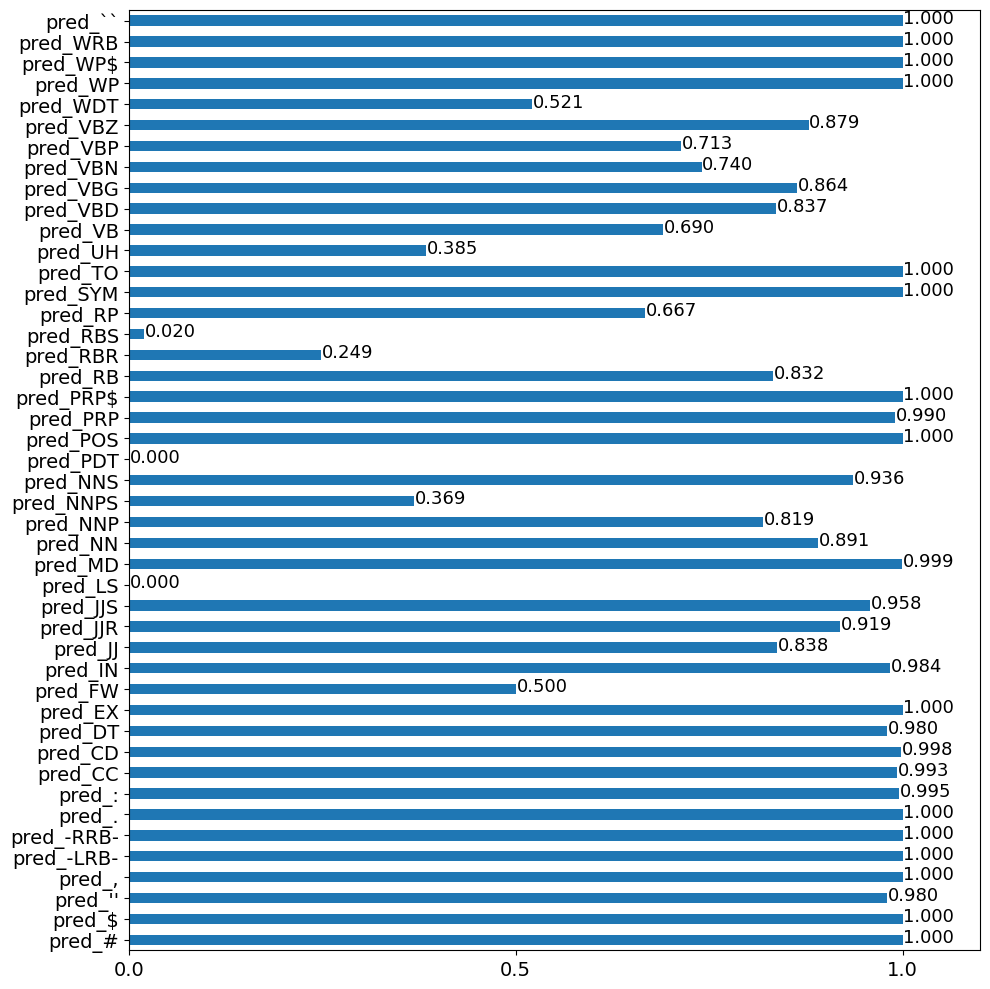
### Recall - Bigrama Validação e Teste (Sem lower)



A esquerda: Recall na validação. A direita: Recall no teste.

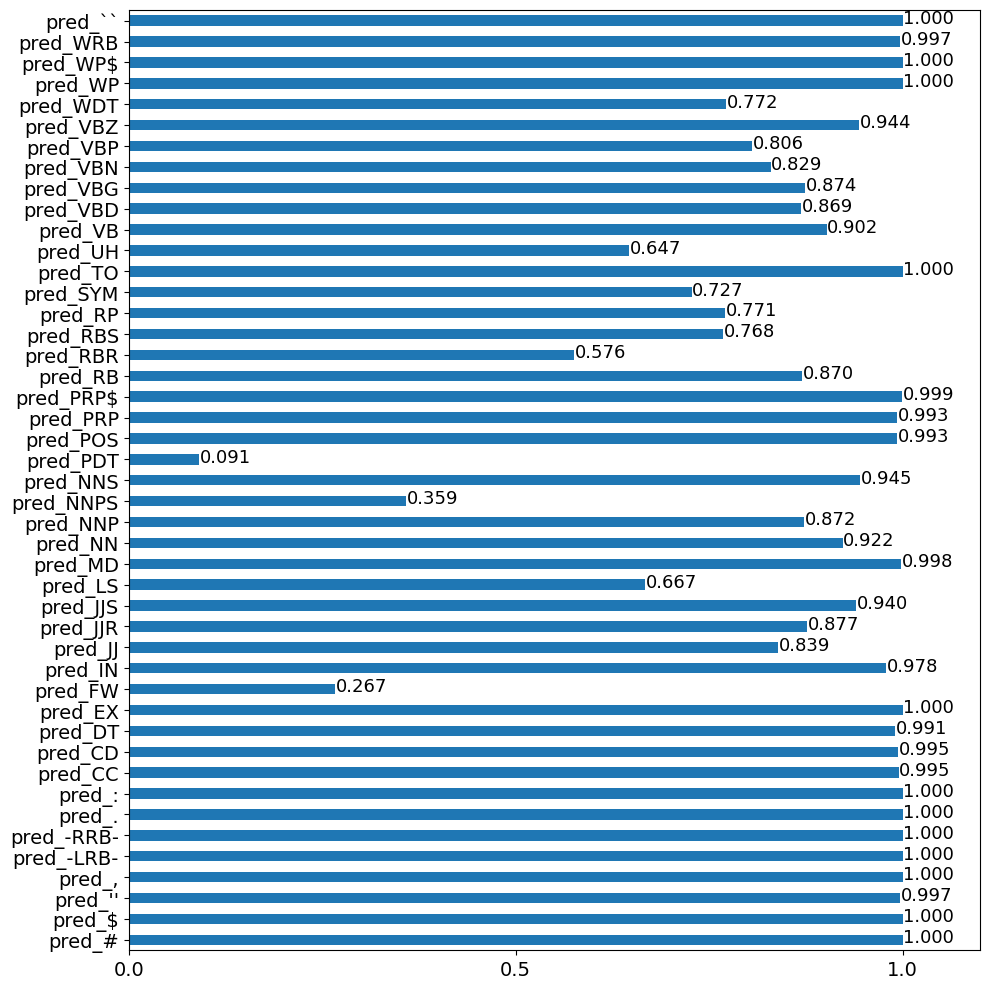
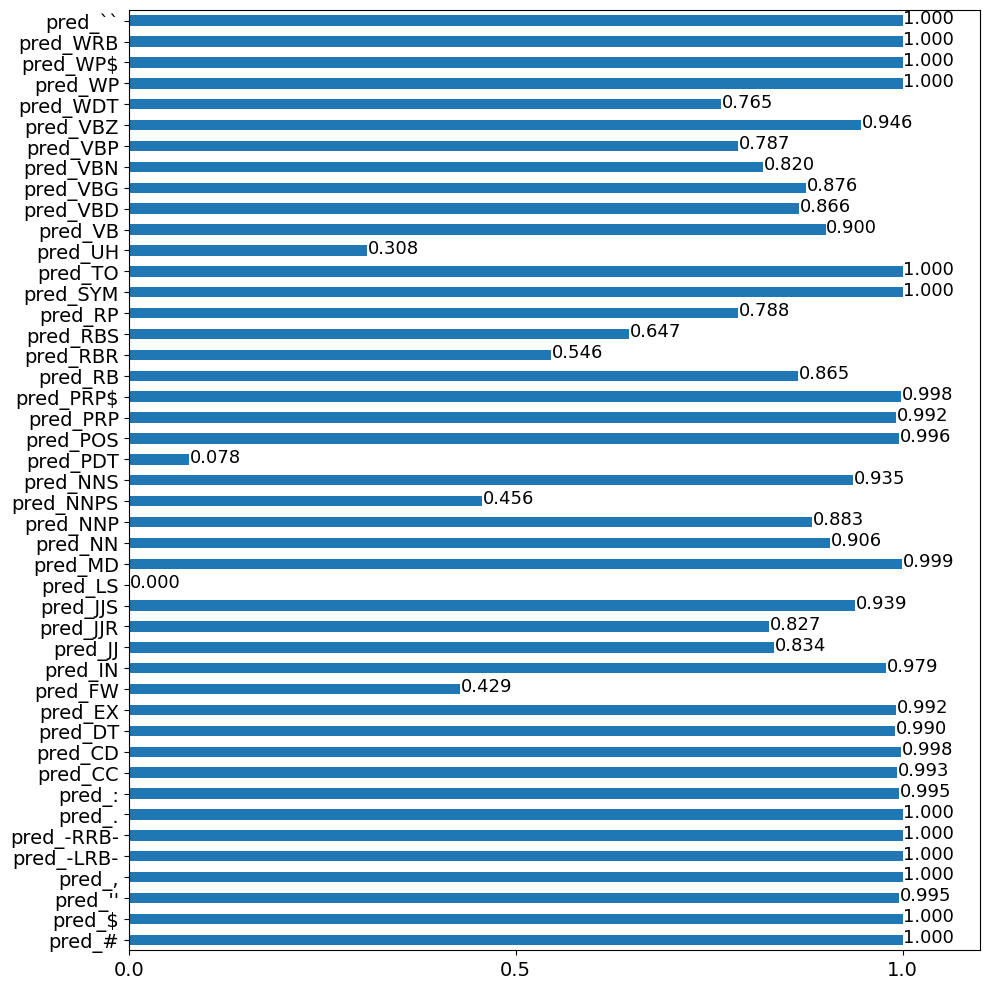
Como podemos perceber, na versão sem o uso do lower como pré-processamento, a versão com bigramas tem um desempenho melhor comparado a versão de unigramas, em diversas tags, tanto o *precision* quanto o *recall* aumentaram.

### Precision - Unigrama Validação e Teste (Com lower)



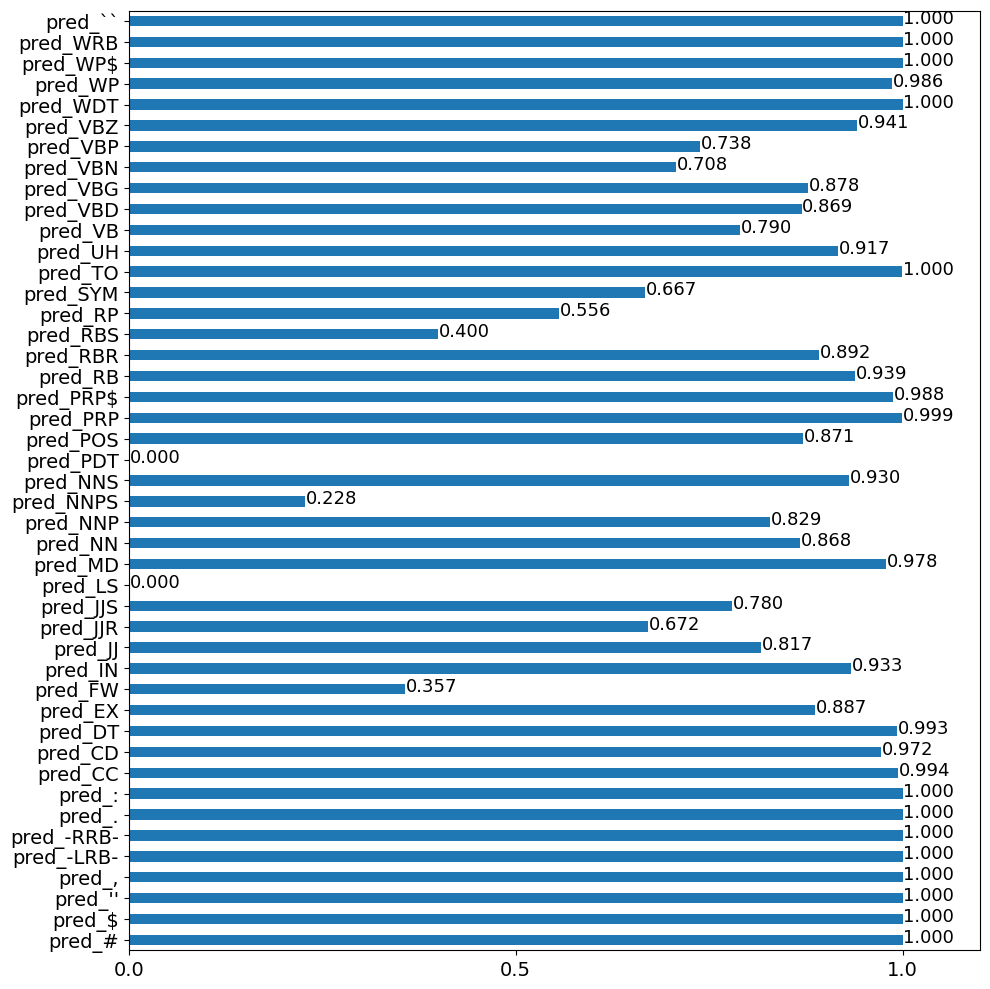
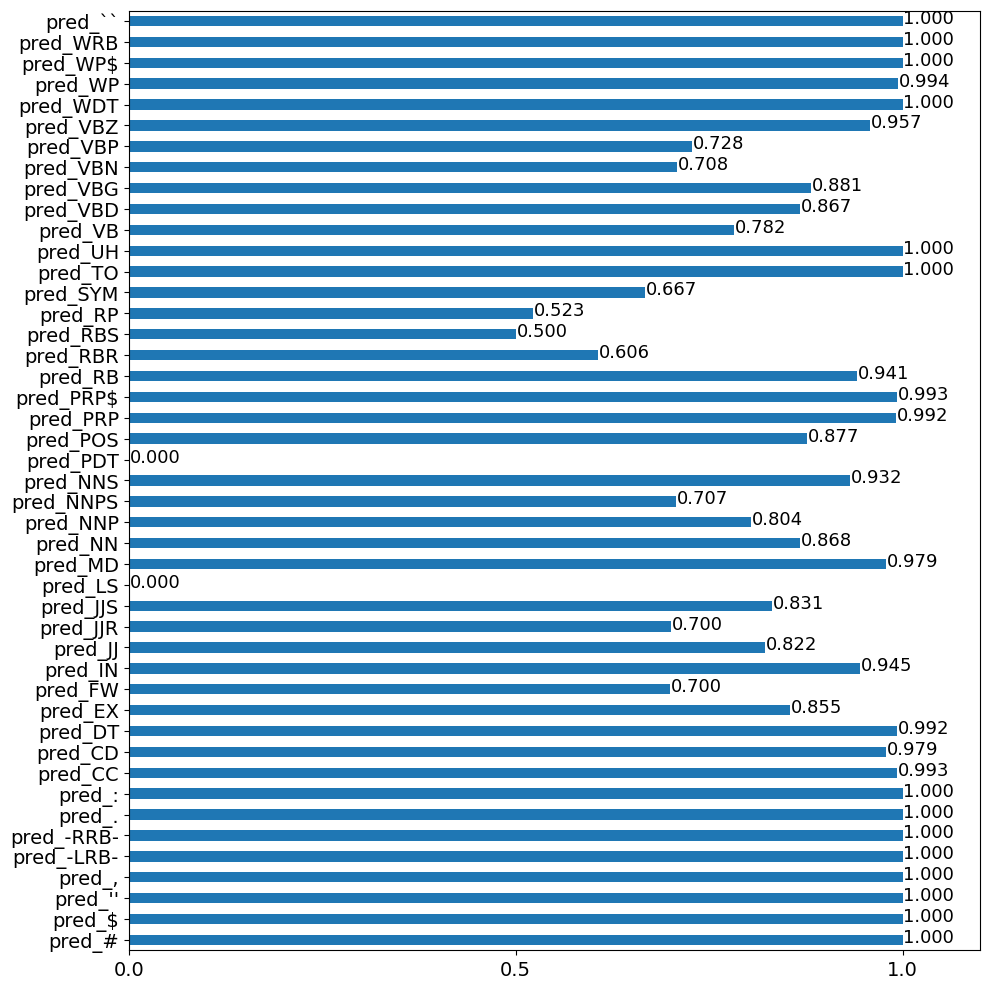
A esquerda: Precision na validação. A direita: Precision no teste.

### Precision - Bigrama Validação e Teste (Com lower)



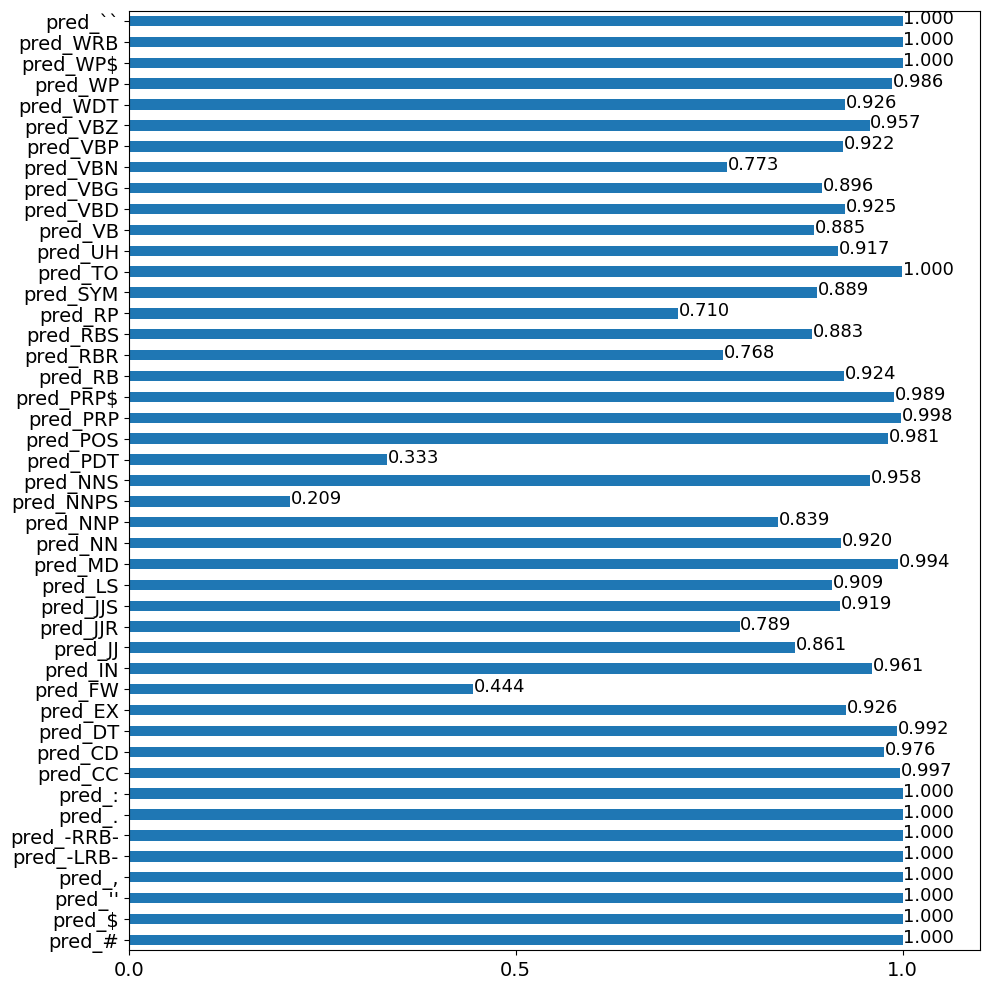
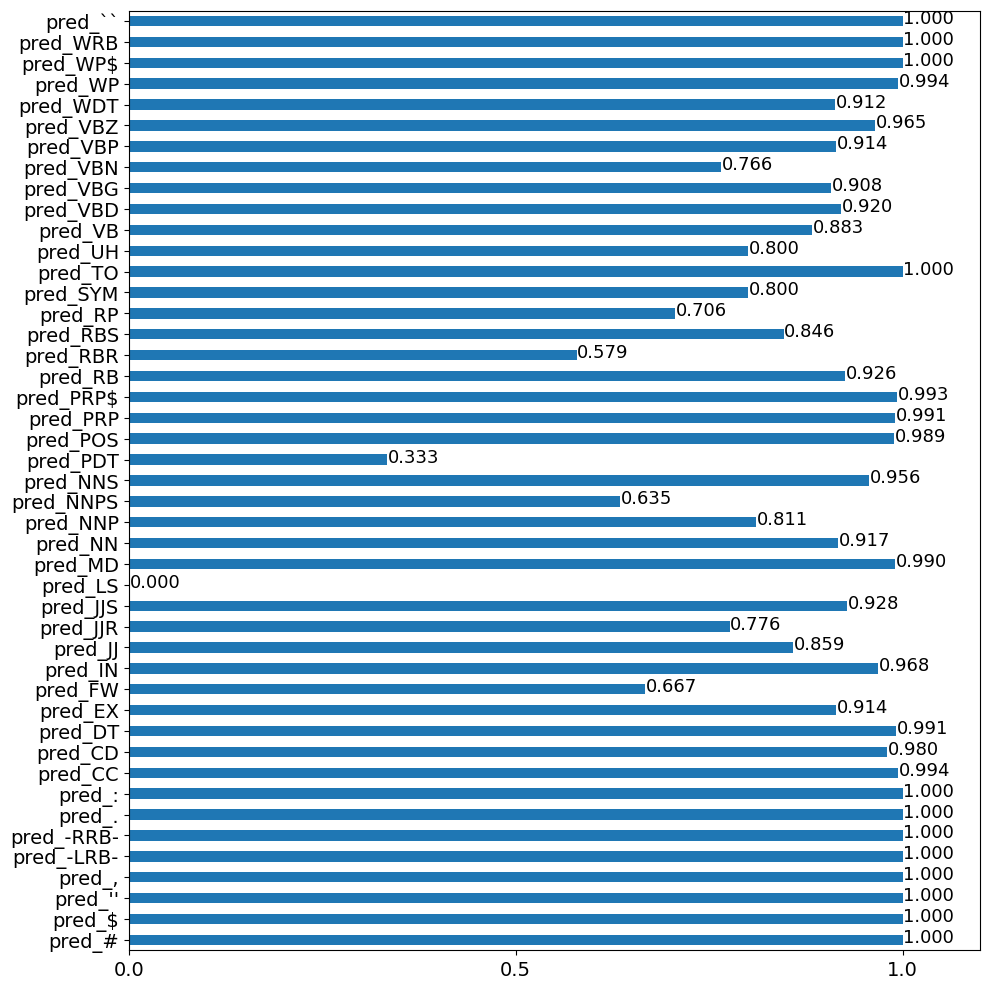
A esquerda: Precision na validação. A direita: Precision no teste.

### Recall - Unigrama Validação e Teste (Com lower)



A esquerda: Recall na validação. A direita: Recall no teste.

### Recall - Bigrama Validação e Teste (Com lower)



A esquerda: Recall na validação. A direita: Recall no teste.

Enquanto que na versão com o uso do lower como pré-processamento, a versão com bigramas continua com um desempenho melhor comparado a versão de unigramas. E ao comparar as versões sem o lower com as versões com o lower, percebemos novamente algumas sutis mudanças tanto no *precision* quanto no *recall.* Algumas tags são melhor classificadas ao utilizar o lower como pré-processamento, enquanto outras são melhor classificadas sem o uso do lower.

Essas e outras métricas podem ser consultadas nos arquivos anexados ao relatório, porém apenas para essas duas, *precision* e *recall*, foram criados esses gráficos. Estão anexados também arquivos que indicam a tag classificada e a tag esperada (a correta) para cada modelo, porém não foram utilizados aqui no relatório.