

Classificadores Naive Bayes informados pelo léxico de sentimentos

Aplicação na análise de críticas de restaurante

Grupo 14

Gustavo Murayama, 21028214

Leonardo Nascimento, 11051613

Matheus Miranda Teles, 11037014

Tiago Suzukayama, 11065914



Expert Systems with Applications

Volume 39, Issue 5, April 2012, Pages 6000-6010



Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews

Hanhoon Kang, Seong Joon Yoo, Dongil Han

Sejong University, Coréia do Sul

117 citações

Cenário

- Números de avaliações de restaurante na internet vem crescendo e assim como a demanda da análise destes dados
- Muitas dessas avaliações não são condizentes ex:
 - Textos positivos com notas negativas
 - Textos negativos com notas positivas

Problema

- A acurácia para classificação positiva tende a ser 10% maior do que a acurácia para classificação negativa, reduzindo a acurácia média
- Léxicos de sentimento generalistas não são bons para a avaliação de críticas de restaurante

Problema

- O SentiWordNet é um léxico de sentimentos generalista
- SentiWordNet não avalia bigramas

Exemplo

- *Delicious* deveria ser avaliada como 100% positiva no contexto
- Avaliada como 75% positiva pelo SentiWordNet
- *Not Delicious* poderia ser avaliado como 100% negativa no contexto

Proposta

- Considerar o léxico de sentimentos para reajustar a função de verossimilhança do classificador Naive Bayes
- Construir um léxico de sentimentos adequado para classificar críticas de restaurante

Resultados

- Melhoria em relação ao classificador SVM
 - até 10,2% na taxa de revocação
 - até 26,2% na taxa de precisão
 - redução de até 28,5% na diferença entre a acurácia das classes
- Melhoria em relação classificador Naive Bayes
 - até 5,6% na taxa de revocação
 - até 1,9% na taxa de precisão
 - redução de até 3,6% na diferença entre a acurácia das classes
- Melhoria significativa quando os atributos são unigramas + bigramas

Implementação

Problema

- O léxico de sentimentos desenvolvido no artigo está indisponível

Proposta

- Considerar o léxico de sentimentos para reajustar a função de verossimilhança do classificador Naive Bayes
- ~~Construir um léxico de sentimentos adequado para classificar críticas de restaurantes~~
- Construir um léxico de sentimentos a partir do SentiWordNet usando o *synset* mais comum

Léxico de sentimientos

Estrutura de dados

$L = \{ \text{"TOKEN": ("POS": 0.6, "NEG": 0.2) } \}$

- **TOKEN:** palavra no formato "word_pos"
- **POS:** positive sentiment value
- **NEG:** negative sentiment value

Procedimento para construção do léxico

- Tokenização de palavras das críticas de restaurante
- Encontrar a classe gramatical das palavras na crítica
- Extrair o significado mais comum da palavra
- Selecionar os sentimentos positivo e negativo desse significado usando o SentiWordNet
- Adicionar no léxico de sentimentos apenas
 - substantivos
 - adjetivos
 - advérbios

Naive Bayes reajustado

Classificador Naive Bayes

$$\text{Class}(d_i) = \arg \max P(c_j) \prod_{i=1}^d P(p_i|c_j)$$

Ingênuo

- Considera que os atributos são independentes entre si
- Não carrega contexto

Proposta

- Informar um pouco de contexto

INB-1

INB-1

$$\text{Class}(d_i) = \arg \max R_1(p_{ij}) P(c_j) \prod_{i=1}^d P(p_i | c_j)$$

$$R_1(p_{ij}) = \frac{\sum_{p_{ij} \in L_j} C(p_{ij})}{\sum_{p_{ij} \in L} C(p_{ij})}$$

```
def _readjustment_log_likelihood(self, X):
    result = []
    for x in X:
        rows, cols = x.nonzero()

        pos = neg = 0
        for col in cols:
            feature = self.feature_names[col]
            score = self.senti_lexicon.get(feature)

            score_delta = score.delta if score is not None else 0

            if score_delta >= 0:
                pos += 1

            if score_delta <= 0:
                neg += 1

        rows_size = rows.size
        if not rows_size:
            pos = neg = rows_size = 1

        neg_ratio = np.max([neg/rows_size, 0.0001])
        pos_ratio = np.max([pos/rows_size, 0.0001])

        result.append([neg_ratio, pos_ratio])

    return np.log(result)
```

Exemplo

An example of a training model.

Pattern	$P(p_i c_{positive})$	$P(p_i c_{negative})$
delicious_V	0.6	0.4
cheap_V	0.4	0.6
noisy_V	0.6	0.4
...		

$$\begin{aligned} \text{likelihood}_{positive} &= R_1 P('delicious_V'|c_{positive}) P('cheap_V'|c_{positive}) \\ &\quad \times P('noisy_V'|c_{positive}) P(c_{positive}) \\ &= (2/3) * 0.6 * 0.4 * 0.6 * 0.5 = 0.048 \end{aligned}$$

$$\begin{aligned} \text{likelihood}_{negative} &= R_1 P('delicious_V'|c_{negative}) P('cheap_V'|c_{negative}) \\ &\quad \times P('noisy_V'|c_{negative}) P(c_{negative}) \\ &= (1/3) * 0.4 * 0.6 * 0.4 * 0.5 = 0.024 \end{aligned}$$

INB-2

INB-2

$$P(p_i|c_j)R_2(p_{ij})$$

$$R_2(p_{ij}) = \begin{cases} \text{if } p_{ij} \in L_j & \alpha = 0.9999 \\ \text{else} & \beta = 0.0001 \end{cases}$$



```
def _readjustment_log_likelihood(self, X):
    result = []
    for x in X:
        _, cols = x.nonzero()

        belonging_cons = np.log(0.9999)
        diverging_cons = np.log(0.0001)

        pos_ratio = neg_ratio = 0
        for col in cols:
            feature = self.feature_names[col]
            score = self.senti_lexicon.get(feature)

            score_delta = score.delta if score is not None else 0

            if score_delta > 0:
                pos_ratio += belonging_cons
                neg_ratio += diverging_cons

            if score_delta < 0:
                pos_ratio += diverging_cons
                neg_ratio += belonging_cons

        result.append([neg_ratio, pos_ratio])

    return result
```


Exemplo

An example of a training model.

Pattern	$P(p_i c_{positive})$	$P(p_i c_{negative})$
delicious_V	0.6	0.4
cheap_V	0.4	0.6
noisy_V	0.6	0.4
...		

$$\begin{aligned} \text{likelihood}_{positive} &= P('delicious_V'|c_{positive})R_2P('cheap_V'|c_{positive})R_2 \\ &= P('noisy_V'|c_{positive})R_2P(c_{positive}) = (0.6 * 0.9999) \\ &\quad * (0.4 * 0.9999) * (0.6 * 0.0001) * 0.5 = 7.199e - 06 \end{aligned}$$

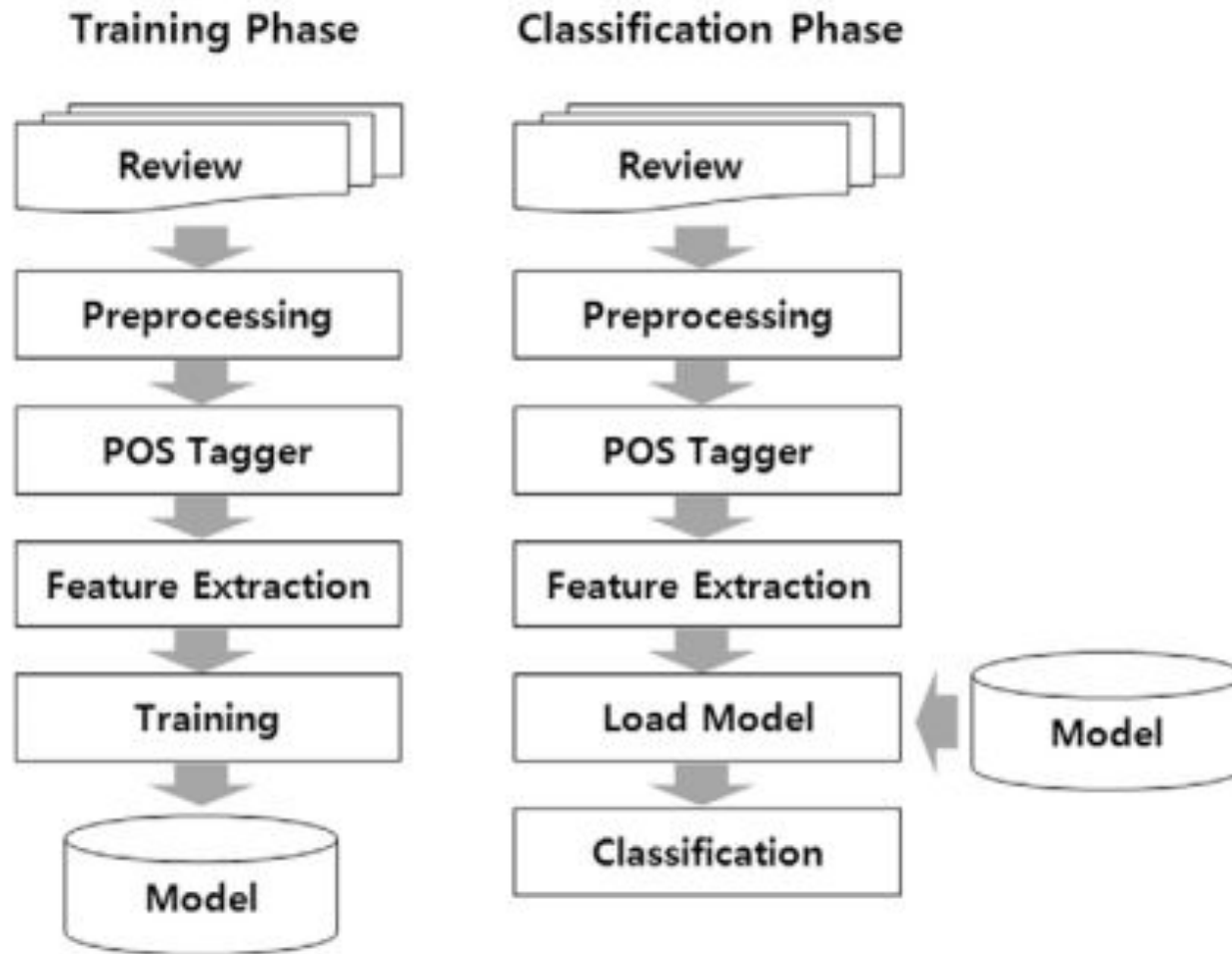
$$\begin{aligned} \text{likelihood}_{negative} &= P('delicious_V'|c_{negative})R_2P('cheap_V'|c_{negative})R_2 \\ &= P('noisy_V'|c_{negative})R_2P(c_{negative}) = (0.4 * 0.0001) \\ &\quad * (0.6 * 0.0001) * (0.4 * 0.9999) * 0.5 = 7.199e - 10 \end{aligned}$$

Conjunto de dados

Restaurant Reviews (Kaggle)

- Conjunto de dados usado em uma [competição](#) criada há 3 anos
- 1.000 reviews de restaurantes
- Conjunto balanceado (500/500)
- Duas colunas
 - Review do restaurante
 - Rótulo

Fluxo



Benchmark

Análise do modelo

- Validação cruzada com 10 folds estratificados
- Avaliando
 - acurácia
 - precisão
 - revocação

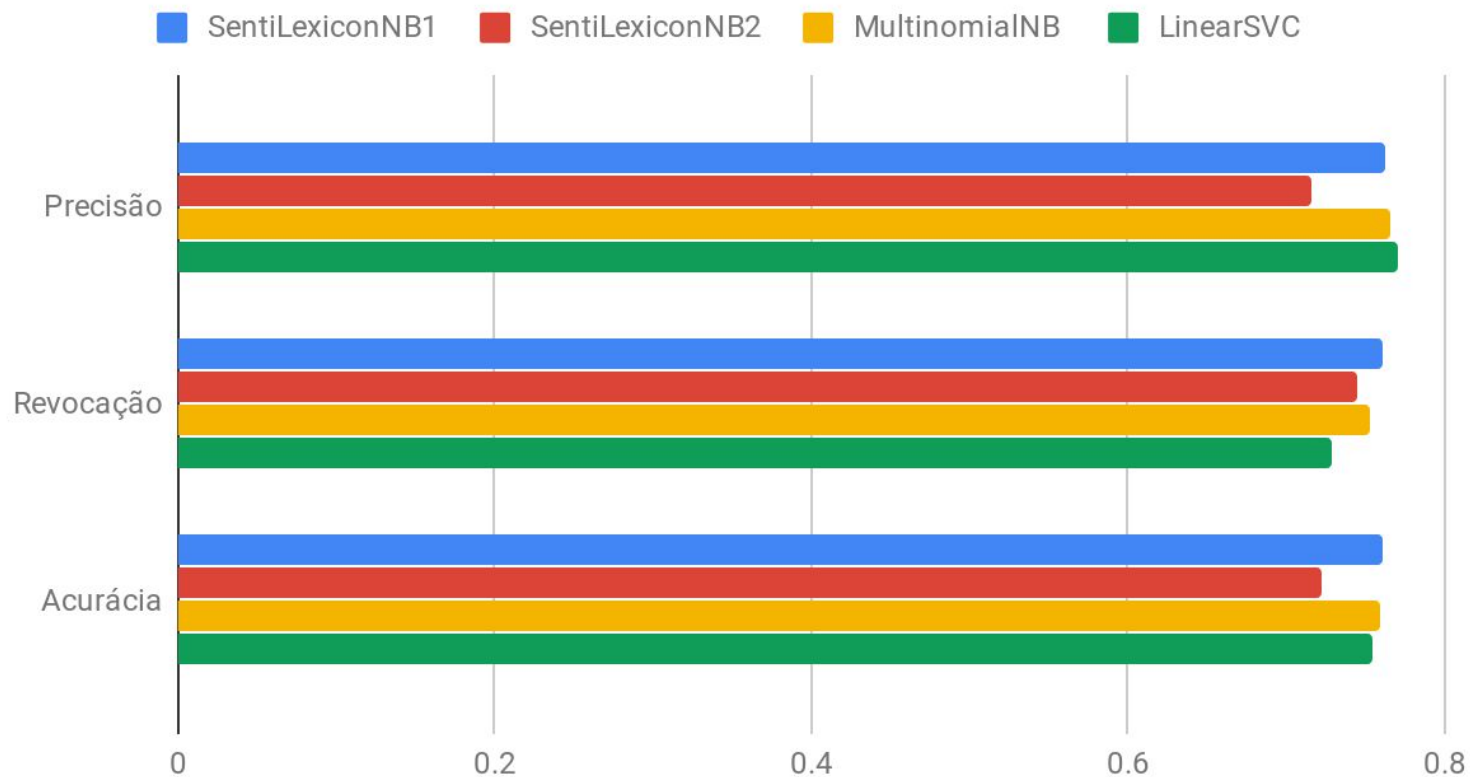
Talk is cheap

[https://repl.it/@leonaascimento/HeartyF
abulousConference](https://repl.it/@leonaascimento/HeartyFabulousConference)

Resultados

Unigramas

Unigramas

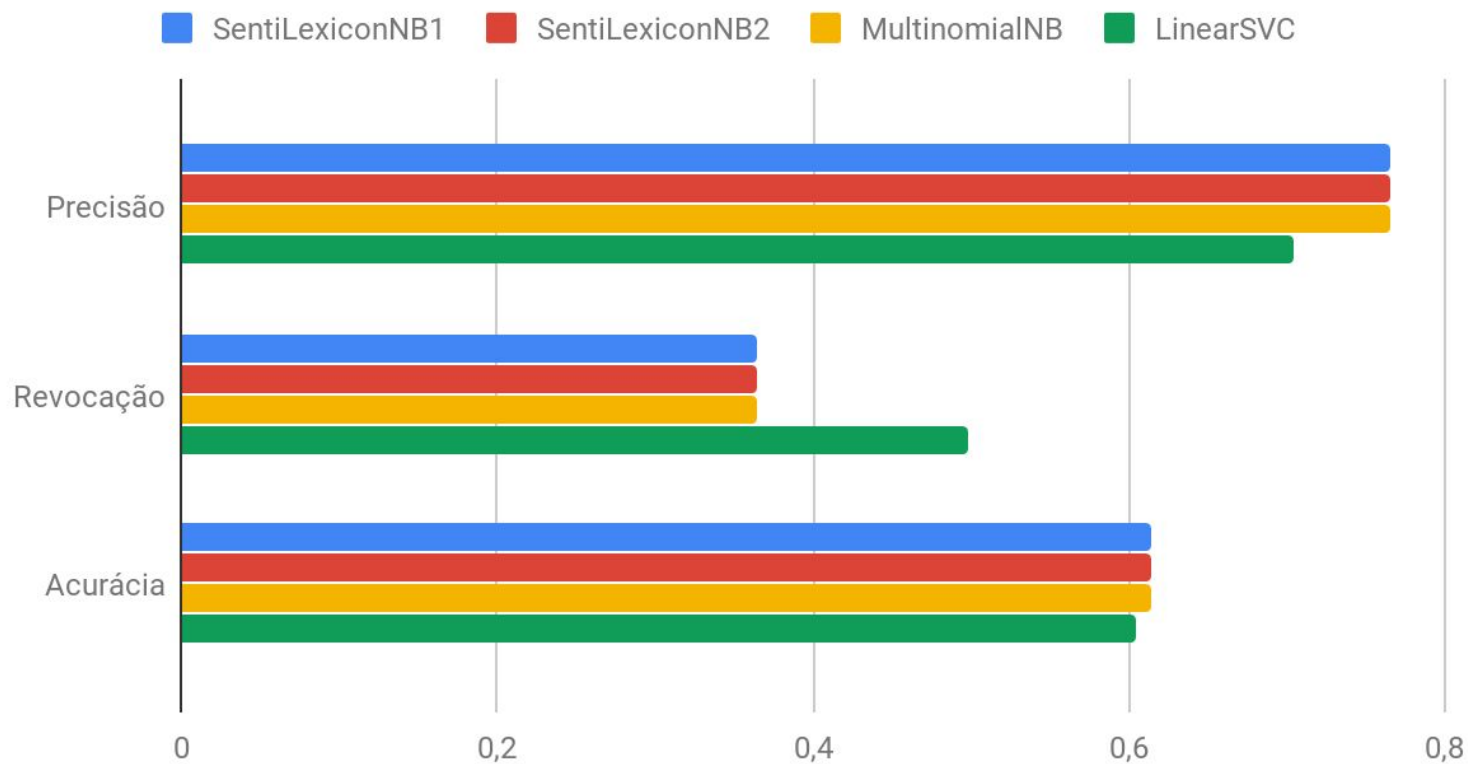


Unigramas

Unigrams				
	INB-1	INB-2	MultinomialNB	LinearSVC
Precisão	0.762 +/- 0.038	0.715 +/- 0.035	0.765 +/- 0.038	0.770 +/- 0.028
Revocação	0.760 +/- 0.053	0.744 +/- 0.039	0.752 +/- 0.059	0.728 +/- 0.038
Acurácia	0.760 +/- 0.033	0.722 +/- 0.026	0.759 +/- 0.031	0.754 +/- 0.017

Bigramas

Bigramas

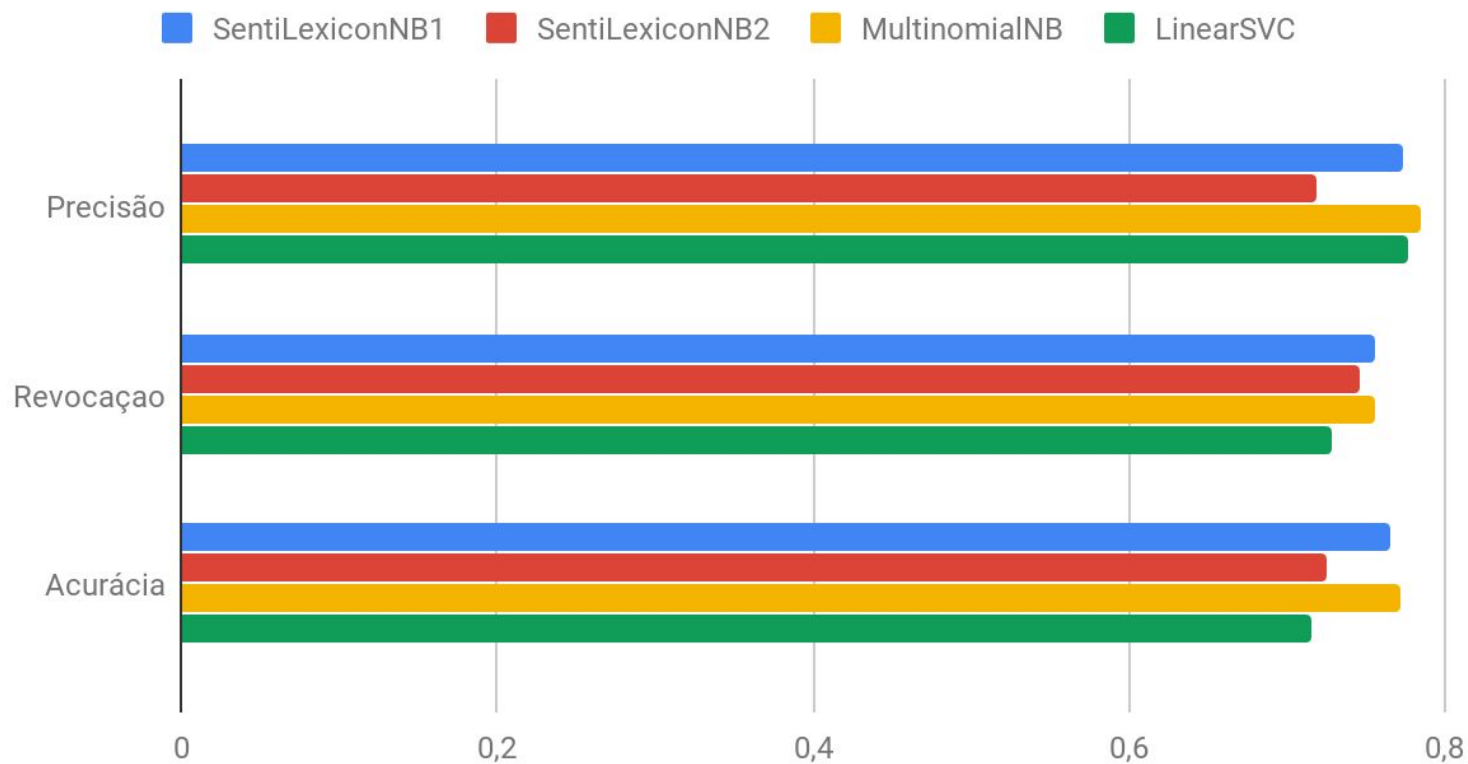


Bigramas

Bigrams				
	INB-1	INB-2	MultinomialNB	LinearSVC
Precisão	0.766 +/- 0.094	0.766 +/- 0.094	0.766 +/- 0.094	0.705 +/- 0.124
Revocação	0.364 +/- 0.202	0.364 +/- 0.202	0.364 +/- 0.202	0.498 +/- 0.305
Acurácia	0.614 +/- 0.035	0.614 +/- 0.035	0.614 +/- 0.035	0.604 +/- 0.036

Unigramas + Bigramas

Unigramas + Bigramas



Unigramas + Bigramas

Unigrams + Bigrams				
	INB-1	INB-2	MultinomialNB	LinearSVC
Precisão	0.773 +/- 0.030	0.719 +/- 0.030	0.784 +/- 0.047	0.776 +/- 0.035
Revocação	0.756 +/- 0.053	0.746 +/- 0.040	0.756 +/- 0.052	0.716 +/- 0.056
Acurácia	0.766 +/- 0.026	0.726 +/- 0.021	0.772 +/- 0.033	0.753 +/- 0.018

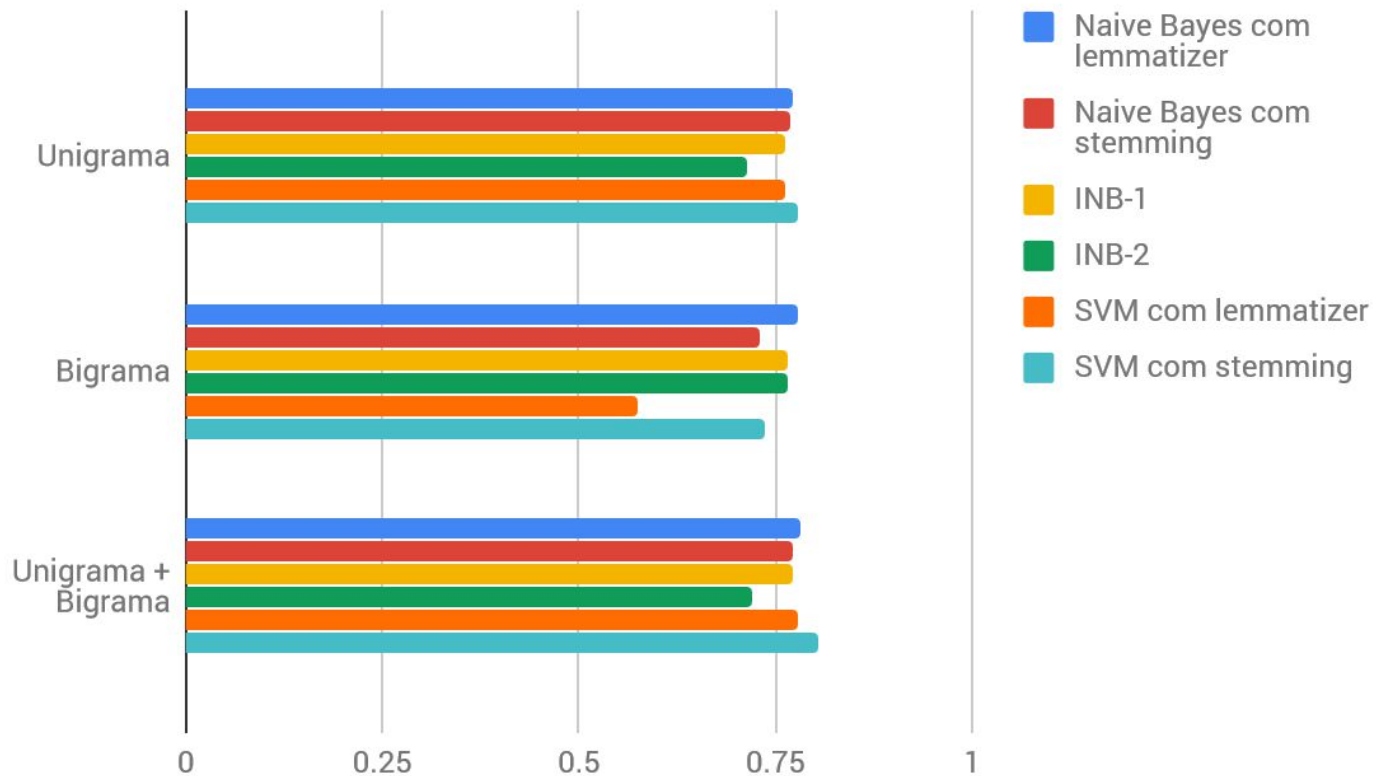
O que poderia ter sido feito melhor?

- Criar o próprio léxico de sentimentos com mais cuidado
- Validar a acurácia, revocação e precisão para cada classe e verificar qual a diferença entre as taxas
- Trabalhar com conjuntos de dados desbalanceados
- Comparar o desempenho com outros conjuntos de dados
- Comparar o desempenho com outros métodos de pré-processamento

Obrigado!

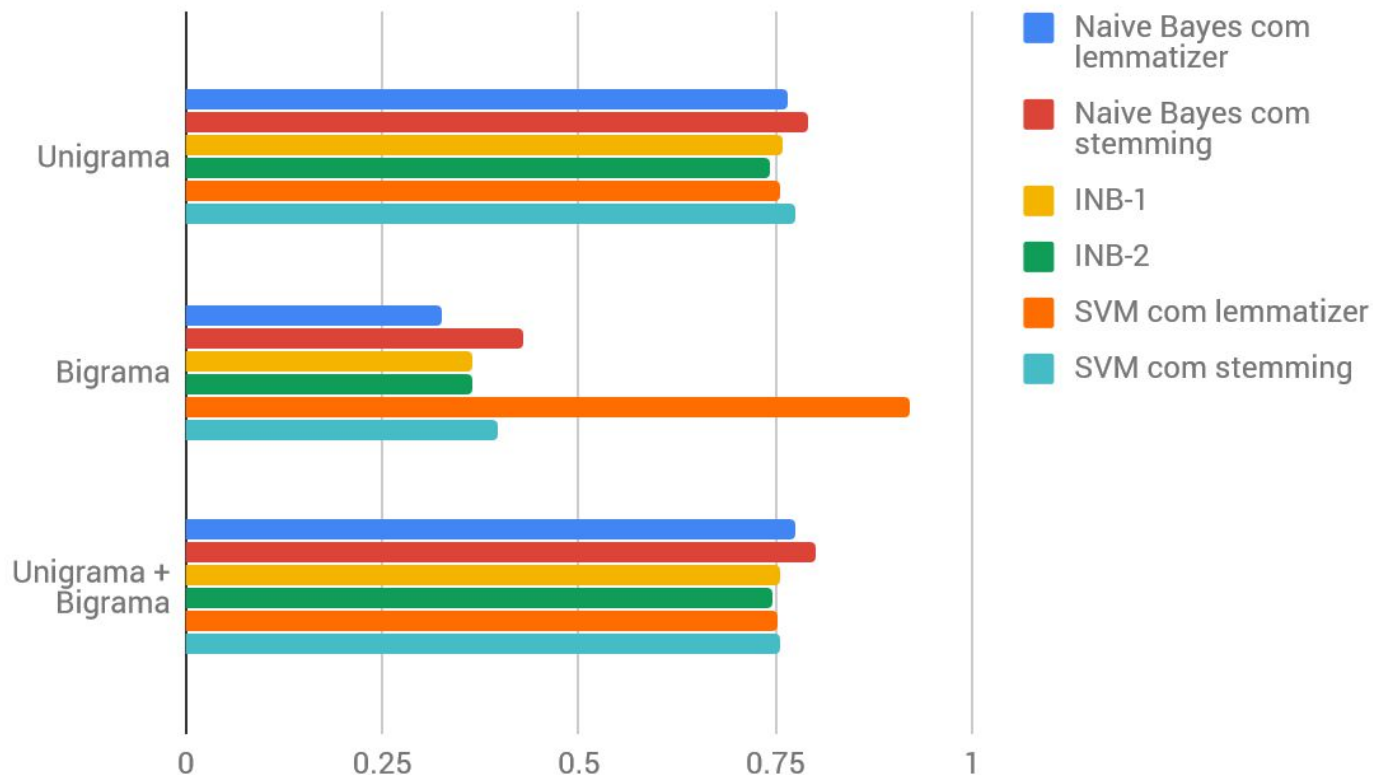
Naive Bayes, INB e SVM

Precisão



Naive Bayes, INB e SVM

Revocação



Naive Bayes, INB e SVM

Acurácia

