**UNIVERSIDADE REGIONAL DE BLUMENAU**

**CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA**

**CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**Disciplina: Processamento de Linguagem Natural**

**Unidade 4 – Análise Sintática, Modelagem e Extração de Informações**

**Data de entrega: 20/05**

**Equipe:** Felipe Bona, João Martinho

A partir dos dados textuais já coletados, cada equipe deverá apresentar e documentar as seguintes tarefas:

1. **Extração de entidades** (PER, ORG, LOC) e/ou **atributos linguísticos** (NOUN, VERB, ADJ etc.), apresentando uma análise estatística básica (entidades mais frequentes, frequência ao longo do tempo etc.), com comentários sobre a importância (ou não) disso para entender a base de dados **(3 pontos).**

Implementação no código:

def analyze\_entities(self, text, verses\_text):

# Dicionário de entidades pré-definidas

entity\_dict = {

'PESSOA': ['Deus', 'Adão', 'Eva', 'Noé', 'Moisés', 'Jesus', 'Paulo',...],

'LOCAL': ['terra', 'céus', 'Egito', 'Israel', 'Jerusalém',...],

'ORGANIZAÇÃO': ['Israel', 'Igreja', 'Sacerdotes', 'Levitas',...]

}

# Dicionário de categorias gramaticais

pos\_patterns = {

'SUBSTANTIVO': r'\b(terra|céus|luz|águas|homem|...)\b',

'VERBO': r'\b(criou|disse|fez|viu|chamou|...)\b',

'ADJETIVO': r'\b(bom|santo|justo|sábio|eterno|...)\b'

}

Análise estatística:

* O código calcula frequência de cada entidade e atributo linguístico;
* Mostra distribuição por versículo (ex: "Versículo 1: PESSOA=2, LOCAL=1, ORGANIZAÇÃO=0"), apresenta totais gerais de cada categoria;
* Relevância para textos bíblicos;
* Identificação de personagens centrais (Deus, Jesus) ajuda na análise narrativa;
* Locais como "Egito" e "Jerusalém" contextualizam eventos históricos;
* Atributos como "criou" (VERBO) e "santo" (ADJETIVO) revelam aspectos teológicos.

1. Aplicação e avaliação qualitativa de abordagens extrativas **– sumarização abstrativa, sumarização extrativa e extração de palavras-chave**. Incluir comentários sobre aplicabilidade e relevância para o conjunto de dados. **(3 pontos)**

def generate\_summaries(self, text, num\_verses):

parser = PlaintextParser.from\_string(text, Tokenizer("portuguese"))

num\_sentences = max(3, num\_verses // 5)

# LexRank

lex\_rank = LexRankSummarizer()

lex\_summary = lex\_rank(parser.document, num\_sentences)

# LSA

lsa\_summarizer = LsaSummarizer()

lsa\_summary = lsa\_summarizer(parser.document, num\_sentences)

Extração de Palavras-chave:

def extract\_keywords(self, verses\_text):

vectorizer = TfidfVectorizer(stop\_words=list(stopwords), max\_features=100)

X = vectorizer.fit\_transform(verses\_text)

feature\_names = vectorizer.get\_feature\_names\_out()

tfidf\_scores = X.max(axis=0).toarray()[0]

return sorted(zip(feature\_names, tfidf\_scores), key=lambda x: x[1], reverse=True)

Avaliação qualitativa:

* LexRank é melhor para destacar eventos narrativos centrais;
* LSA identifica melhor temas latentes e relações conceituais;
* TF-IDF é eficaz para identificar termos centrais, mas pode perder contexto.

1. Proposta e/ou aplicação de **classificação de textos por aprendizado supervisionado**, incluindo avaliação quantitativa e qualitativa das métricas (precisão, acurácia, recall etc.), estratégias para rotulação dos dados e resultados obtidos/esperados. **(4 pontos)**

def classify\_text(self, verses\_text):

# Dados de treino rotulados manualmente

sample\_texts = [

("Deus criou os céus e a terra", "positive"),

("A terra era sem forma e vazia", "negative"),

...

]

# Vetorização e treinamento

vectorizer = TfidfVectorizer(max\_features=1000, stop\_words=...)

X = vectorizer.fit\_transform(texts)

y = labels

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

# Modelo Naive Bayes

classifier = MultinomialNB(alpha=1.0)

classifier.fit(X\_train, y\_train)

# Avaliação

cv\_scores = cross\_val\_score(classifier, X, y, cv=5)

y\_pred = classifier.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

Métricas e avaliação:

* Acurácia: medida de acertos gerais;
* Precisão: capacidade de não classificar incorretamente como positivo;
* Recall: capacidade de encontrar todos os positivos;
* F1-Score: média harmônica entre precisão e recall;
* Validação cruzada: mostra robustez do modelo.

Estratégia de rotulação:

* Baseada em contexto teológico;
* Positivo: criação, bênçãos, esperança;
* Negativo: julgamento, pecado, destruição;
* Neutro: descrições factuais.