Expressões Regulares, Tokenização e Distância de Edição

1. Introdução

O capítulo 2 do livro *Speech and Language Processing* aborda ferramentas fundamentais no processamento de linguagem natural (PLN), com foco em expressões regulares, normalização de texto, segmentação de palavras e sentenças, stemming, lematização e distância mínima de edição. Esses elementos formam a base para tarefas posteriores de compreensão e geração de linguagem por máquinas.

2. Expressões Regulares (Regex)

2.1 Conceito e História

Expressões regulares são uma linguagem formal desenvolvida originalmente por matemáticos como Stephen Kleene na década de 1950 para descrever padrões em cadeias de caracteres. No contexto computacional, elas se tornaram ferramentas poderosas para busca, substituição e extração de padrões em texto.

2.2 Sintaxe e Elementos Principais

- **Concatenação**: "abc" casa com a sequência exata.
- **Disjunção (alternância)**: [abc] ou a|b|c casa com qualquer caractere listado.
- **Quantificadores**:
    - *: zero ou mais ocorrências
    - +: uma ou mais ocorrências
    - {n,m}: entre n e m ocorrências
- **Âncoras**:
    - ^: início da string
    - $: fim da string
- **Agrupadores e precedência**: (abc) cria grupos e define ordem de precedência.

2.3 Aplicações Comuns

- Validação de formatos (e-mails, números de telefone)
- Extração de informações (nomes, datas)
- Substituição e limpeza de dados em tarefas de pré-processamento

3. Tokenização e Normalização

3.1 Tokenização de Palavras

A tokenização divide o texto em unidades menores (tokens), geralmente palavras. Um exemplo simples seria dividir a frase:

"A inteligência artificial é fascinante."

em:

["A", "inteligência", "artificial", "é", "fascinante", "."]

Em muitos idiomas, essa tarefa é complexa por causa de:

- Contrações ("não é" → "não", "é")
- Palavras compostas
- Idiomas sem separação por espaço (ex.: chinês)

3.2 Segmentação de Sentenças

A segmentação de sentenças é o processo de identificar onde termina uma sentença. Desafios incluem:

- Ponto final em abreviações (ex: "Dr.", "Sra.")
- Pontuação ambígua
- Quebras de linha

Exemplo:

"Olá! Como vai? Hoje é sexta-feira." -> Segmentado em três sentenças.

4. Normalização de Texto

4.1 Conceito

A normalização visa padronizar variações linguísticas para facilitar o processamento. Pode incluir:

- Conversão para minúsculas
- Remoção de acentos
- Substituição de contrações
- Correção ortográfica

4.2 Exemplo Prático

Texto original:

"R$16,98 é o preço. 6^2 metros."

Normalizado:

"16.98 reais é o preço. 6 ao quadrado metros ao quadrado."

Transformações como:

- R$16,98 → 16.98 reais
- 6^2 → 6 ao quadrado
- m^2 → metros ao quadrado

5. Stemming e Lematização

5.1 Stemming

O stemming remove sufixos para reduzir palavras à sua raiz. É uma abordagem heurística. Exemplo com o algoritmo de Porter:

- “running”, “runner” → “run”
- “amando”, “amava” → “am”

Vantagens:

- Simples e rápido

Desvantagens:

- Pode gerar palavras que não existem

5.2 Lematização

A lematização utiliza dicionários e regras gramaticais para reduzir a palavra à sua forma canônica (o lema). Exemplo:

- “melhores” → “bom”
- “correndo” → “correr”

Vantagens:

- Linguisticamente correta
- Preserva o significado

Desvantagens:

- Mais complexa e lenta

6. Distância Mínima de Edição

6.1 Conceito

A distância mínima de edição (ou distância de Levenshtein) é o número mínimo de operações necessárias para transformar uma string em outra. Operações incluem:

- Inserção
- Remoção
- Substituição

6.2 Aplicações

- Correção ortográfica
- Comparação de nomes
- Detecção de plágio

6.3 Exemplo

Palavras: **gato** e **rato**

Operações:

- Substituir “g” por “r” → 1 operação

Distância = 1

6.4 Algoritmo

O cálculo é feito por programação dinâmica, criando uma matriz de custos entre os caracteres das duas strings e preenchendo-a linha por linha.

7. Conclusão

O processamento de linguagem natural começa com a capacidade de interpretar e manipular o texto bruto de forma estruturada e significativa. Nesse capítulo, exploramos ferramentas fundamentais para essa missão — das expressões regulares à distância de edição — que fornecem a base para praticamente todas as aplicações modernas de PLN, como chatbots, mecanismos de busca, tradutores automáticos e análise de sentimentos.

Através das expressões regulares, aprendemos a identificar padrões com precisão e eficiência. Com a tokenização e a normalização, conseguimos preparar o texto para etapas analíticas mais profundas. O stemming e a lematização nos ajudam a abstrair o conteúdo, reduzindo palavras à sua essência sem perder o significado. E com a distância mínima de edição, temos um mecanismo robusto para medir semelhança textual e corrigir erros.

Mais do que simples técnicas isoladas, esses conceitos representam o primeiro passo no caminho da compreensão computacional da linguagem humana. Dominar essas ferramentas permite não apenas automatizar tarefas repetitivas, mas também abrir portas para análises linguísticas complexas, tornando possível que máquinas entendam — mesmo que ainda de forma limitada — o que dizemos, escrevemos e pensamos.