

**Disclaimer:** *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

## 12.1 Introdução à Análise Estatística de Texto

Neste tema vamos focar na forma de media mais difundida e presente: *texto*. O texto está em todo lado, na linguagem falada, livros, websites, etc. Não é surpresa que o sucesso dos grandes motores de busca seja baseado na análise adequada de texto. Derivar ‘significado’ do texto está longe de ser trivial, na verdade é uma tarefa muito difícil. Uma das frases que discutimos nas aulas foi ‘semântica a partir de lexis’: o significado no texto em sistemas inteligentes é criado por distribuições de palavras numa linguagem específica, seguindo regras gramaticais muito específicas e diversas. Estes padrões linguísticos são ainda mais refinados (e complicam ainda mais o problema ‘semântica a partir de lexis’) quando consideramos códigos culturais, abreviaturas, metáforas, analogias, ironia, referências específicas, e a forma como as pessoas escrevem online.

## 12.2 O Pipeline Bag-of-Words (Saco de Palavras)

Durante as aulas falamos brevemente sobre a história de linguística computacional, particularmente como o seu foco inicial era o conseguir receber, por exemplo, uma frase, e produzir uma saída na qual os padrões gramaticais estavam claramente identificados. A esperança era que tal representação, em conjunto com regras lógicas poderia resolver muito do nosso problema ‘semântica a partir de lexis’. Na prática, os linguistas computacionais rapidamente encontraram um grande problema: a comunicação humana segue regras (gramaticais, linguísticas, culturais) mas quase sempre com alguns ‘erros’ ou divergências do que é, em termos de linguística formal, correto. Mas principalmente, a anotação de estruturas gramaticais e relacionamento a partir de regras lógicas não permitia ao sistema inteligente inferir nada sobre o significado dos símbolos que estava a processar. Numa vertente mais ‘data science’ outra escola dentro de IA, preocupados com problemas tais como os motores de busca, começaram a pensar no problema ‘semântica a partir de lexis’ com uma abordagem mais estatística. O ponto de partida foi o conceito de ‘Corpus’: um conjunto de documentos de texto, tipicamente coerentes em termos temáticos. Nas aulas usamos um corpus de receitas de cozinha. Corpora multi-temáticas são também corpus (como o conjunto de paginas na web) mas estas são instâncias muito mais desafiantes, por isso começamos com corpora mais simples (mono-temáticas).

O ciclo de processamento estatístico de texto tem quatro etapas, ilustradas na Figura 12.1. O ponto de partida deste processo é a disponibilidade de um **corpus**, que nada mais é do que uma coleção de documentos de texto. Por exemplo, podemos ter um corpus sobre debates políticos, ou receitas de culinária, as notícias transmitidas por uma determinada agência num período de tempo, etc. Corpora (plural de corpus) frequentemente reúnem muitos documentos sobre um determinado tema, e o nosso objetivo é encontrar os diferentes tópicos que compõem esse tema.

Uma vez que temos o nosso corpus, a primeira etapa está relacionada com **pré-processar os documentos**. A ideia principal aqui é remover o ‘ruído’ na forma de texto que pouco tem a dizer sobre o que um documento está a falar. A análise estatística de texto (STA) está preocupada com as distribuições de palavras. Isto

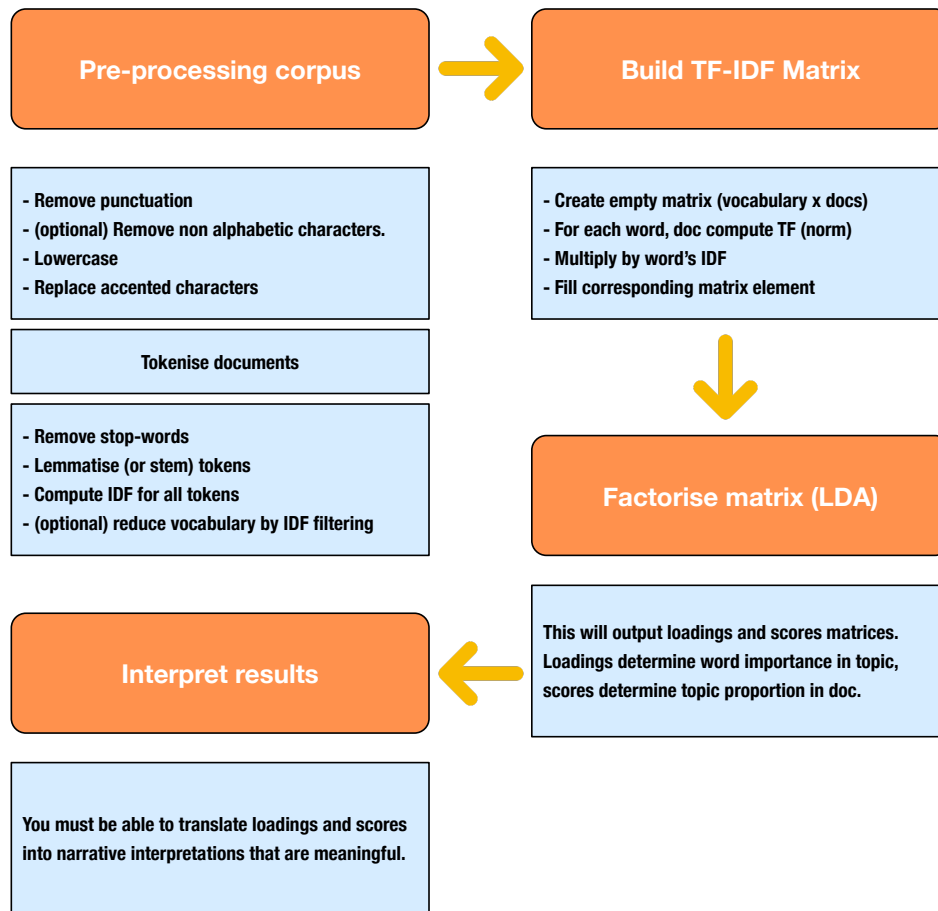


Figure 12.1: As quatro etapas principais no processamento estatístico de texto.

é assim no sentido de que um documento que contém, por exemplo, muitas cópias das palavras ‘forno’, ‘cozinhar’ e ‘cebola’ provavelmente é sobre receitas/comida. A STA não se preocupa com regras gramaticais de nenhum tipo. Por esta razão, há muitas palavras e pontuação que não são importantes para a STA (e de fato são eliminados como veremos mais tarde). Outros métodos de análise de texto no campo do processamento profundo de linguagem natural (Deep NLP) estão interessados (e utilizam) regras gramaticais, mas não estudamos Deep NLP nesta cadeira.

### 12.3 Pré-processamento

Para limpar os documentos no nosso corpus, carregamos iterativamente cada documento no nosso programa de computador, (1) removendo todos os caracteres que não interessam, particularmente pontuação (mas talvez também números, emoji e outros dependendo dos objetivos de análise do corpus); (2) alterando todo o texto para minúsculo. Às vezes (especialmente quando se utilizam dicionários ao longo do pipeline de análise) podemos optar por manter todas as maiúsculas apropriadas, mas em geral o texto é todo alterado para minúsculas padrão; (3) substituímos caracteres com acento pela sua variante sem acento. Neste momento implementamos um passo intermédio antes de prosseguir para a segunda parte do pré-processamento: (4)

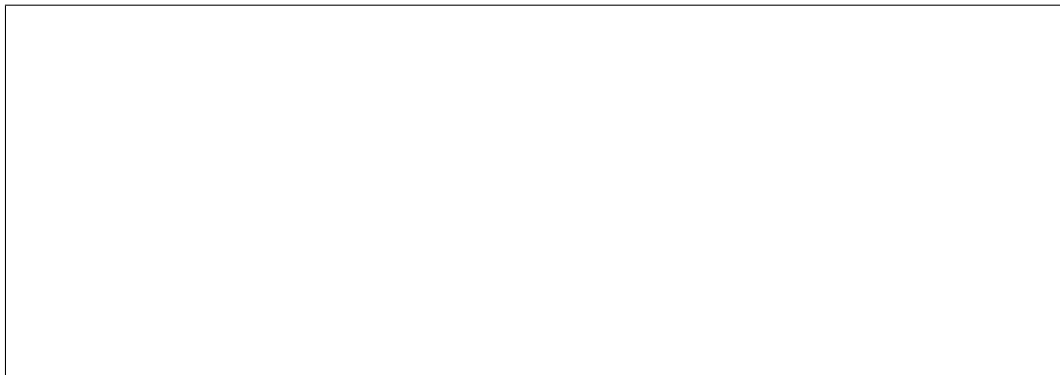
tokenização, isto é a cadeia de caracteres que representa o documento, é substituída por uma lista das suas palavras componentes. Na segunda parte do pré-processamento, (5) removemos palavras com função estritamente gramatical (stop words). Por exemplo, em inglês, termos como ‘in’, ‘at’, ‘to’, etc., são todos eliminados porque não nos fornecem informações sobre o que o documento trata. Este passo já reduz significativamente o número de tokens que podem aparecer nos documentos do nosso corpus. O ultimo passo tem o objetivo de standardizar variantes léxicas de um mesmo conceito para uma única palavra raiz. Este passo é chamado **lematização ou stemming** (). Desta forma, se fizermos lematização, os tokens mapeado, mapeando, mapeador e mapas são todos substituídos pelo único lema ‘map’. Isso comprime claramente ainda mais a representação dos nossos documentos. O stemming substitui as mesmas palavras pela raiz etimológica (no caso dos mapas, a raiz seria ‘map’).

Neste ponto, e em preparação para a próxima etapa de processamento, é importante derivar o **dicionário** de tokens no corpus. Esta é simplesmente uma lista de todos os termos que aparecem pelo menos uma vez em pelo menos um documento dentro do nosso corpus, é por isso que o chamamos de dicionário universal.

Considere o seguinte corpus reduzido composto por três documentos:

$$\begin{aligned} D_1 &= \{cebola, alho, alho, morango\} \\ D_2 &= \{tomate, manteiga, tomate, pra, banana\} \\ D_3 &= \{leite, farinha, cebola, cebola, tomate\} \end{aligned}$$

Qual é o dicionário universal correspondente de termos?



Note que para a implementação desta etapa de limpeza, podemos encontrar bibliotecas e recursos, como dicionários de stop words e lematizadores, disponíveis para as linguagens de programação mais utilizadas, como por exemplo Python.

## 12.4 Construção da Matriz TF-IDF

Esta etapa é central para a STA usando a abordagem BoW. Um dos aspectos mais interessantes é que, enquanto começamos com um corpus composto por elementos disjuntos (uma coleção de documentos), acabamos com uma representação única do corpus na qual todos os documentos estão representados com o mesmo formato (vetorial), o qual permite fazer inferências semânticas a partir de relações vetoriais simples.

### 12.4.1 Frequência de Termos: TF

A Frequência de Termos, ou simplesmente TF, é uma quantidade numérica usada para expressar a importância de um termo dentro de um documento. Na sua forma básica, é simplesmente a contagem do número de vezes que um termo aparece num documento. No entanto, aqui calculamos o TF como uma proporção, dividindo esta contagem pelo número total de tokens no documento pré-processado. Isto significa que, por exemplo, o TF do termo alho no Documento 1 acima é  $tf(garlic, D_1) = 2/4 = 1/2$ , enquanto o TF de tomate no Documento 2 é  $tf(tomato, D_2) = 2/5$ .

Calcular o TF desta maneira significa que o valor de TF variará entre  $[0, 1]$  e também que podemos comparar qualquer par de documentos sem nos preocupar com viés causado por um dos documentos ser muito maior do que o outro. Em outras palavras, esta forma de calcular o TF coloca todos os documentos num único padrão numérico.

### 12.4.2 Frequência Inversa de Documentos: IDF

A frequência de termos não conta toda a história de que precisamos para analisar um corpus. O TF relaciona termos e os documentos que os contêm. Mas e a importância dos termos no corpus? Suponha, por exemplo, que no nosso corpus a palavra ‘arroz’ apareça em todos os documentos. Suponha que está na função de bibliotecário que mantém este corpus, e alguém vem à procura dum subconjunto de documentos sobre um determinado tópico do seu corpus. Imagine que este visitante da biblioteca lhe diz ‘arroz’. Quando for até a caixa buscar todos os documentos que contêm essa palavra, voltará com a caixa inteira porque todos os documentos contêm esse termo. O termo ‘arroz’ foi útil para apoiar as necessidades do visitante da biblioteca? Não muito. Na verdade, nada útil.

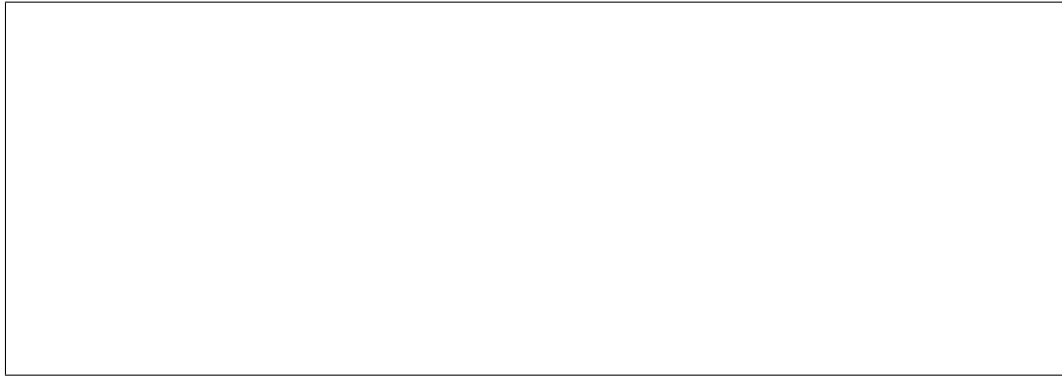
É aqui que a Frequência Inversa de Documentos, ou IDF, é útil. Este número representará a importância de um termo num determinado corpus, calculado usando a seguinte fórmula:

$$IDF_t = \text{Log} \left( \frac{N}{df_t} \right)$$

onde  $t$  refere-se ao termo,  $N$  ao número total de documentos no corpus e  $df_t$  ao número de documentos que contêm o termo  $t$  pelo menos uma vez. Seguindo isso, podemos calcular que o IDF de cebola é

$$IDF_{onion} = \text{Log} \left( \frac{3}{2} \right)$$

Qual é o IDF de alho? E faz sentido calcular o IDF do termo ‘avião’ neste corpus?



### 12.4.3 Frequência de Termos Inversa da Frequência de Documentos: TF.IDF

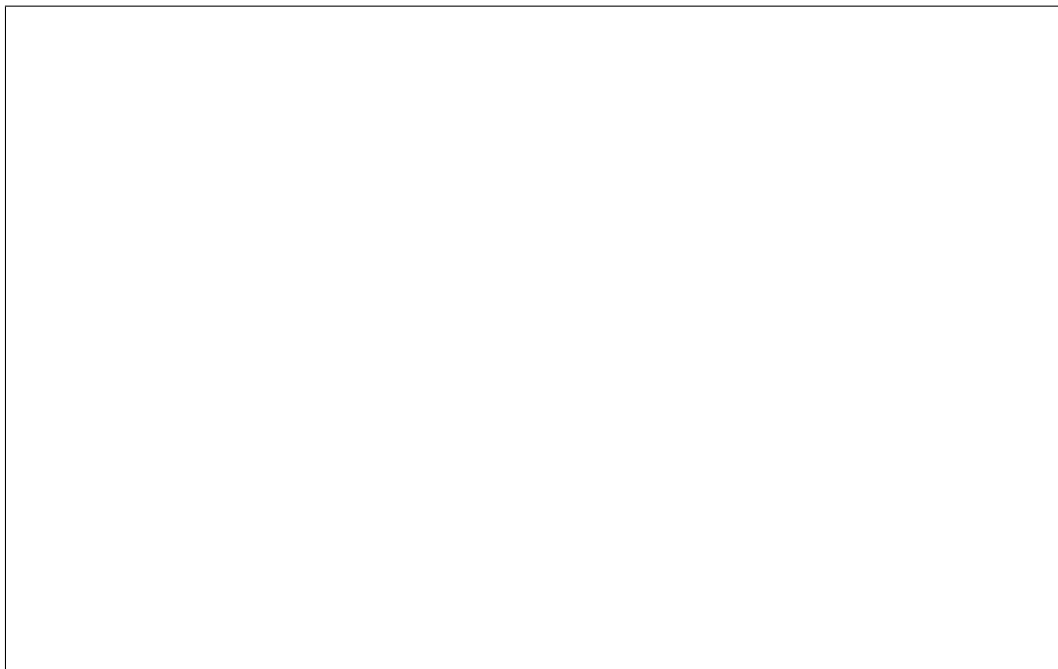
A quantidade final que usaremos para medir a importância de um termo dentro de um documento que pertence a um corpus é a medida padrão TF-IDF, que é simplesmente:

$$\text{TF-IDF}_t^d = TF_t^d \times IDF_t$$

Aqui  $t$  refere-se sempre ao termo, e  $d$  a um documento específico. Qual é o efeito de multiplicar o TF normalizado original pelo IDF? O IDF atua como um modulador. Se o TF for alto, mas o termo estiver em todo o corpus, o IDF será baixo, então o TF é reduzido. Se um TF for médio, mas o IDF for alto, então a sua importância é aumentada.

Voltando ao foco desta seção, agora temos tudo o que precisamos para construir a nossa representação única do corpus como uma matriz. Esta matriz  $S$  tem linhas representando os termos do dicionário universal para o corpus e colunas representando os documentos contidos. Portanto, uma determinada célula  $S_{t,d}$  da matriz conterá o correspondente  $\text{TF-IDF}_t^d$ .

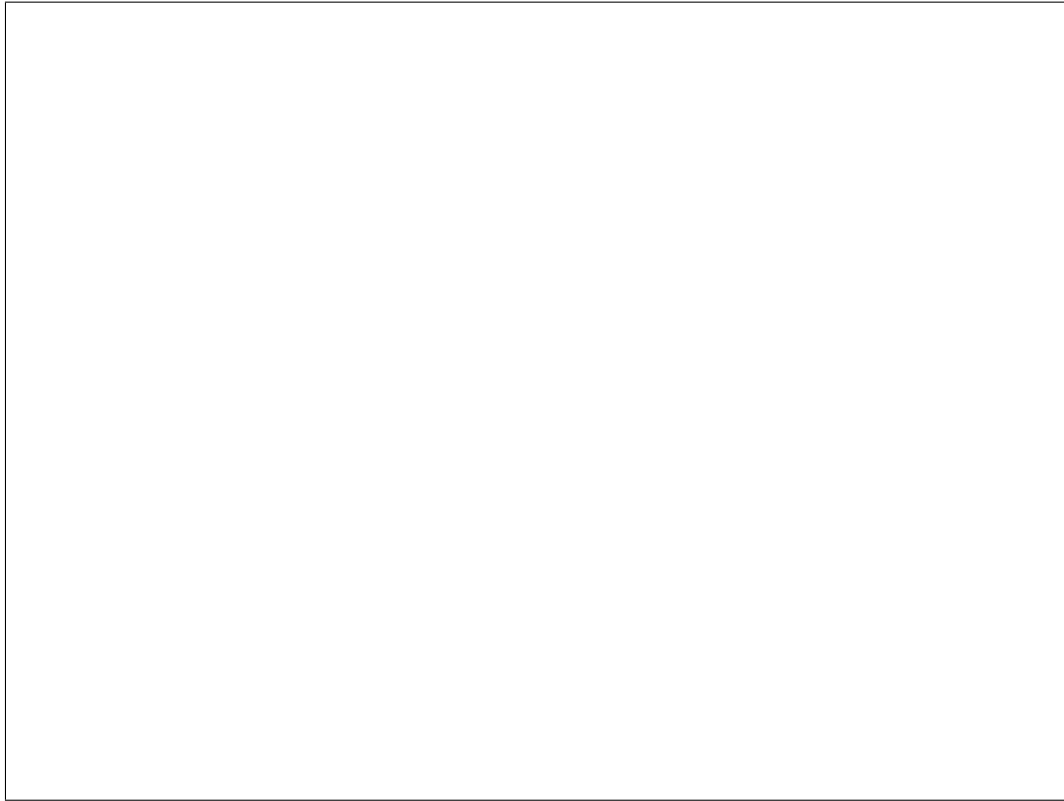
Calcule e valide a matriz  $S$  completa para o corpus de exemplo nesta nota no espaço abaixo:



Uma vez que temos a nossa matriz pronta, podemos começar a perceber o que esta representa. Em outras palavras, podemos começar a fazer algumas coisas inteligentes. É muito importante perceber que cada coluna da matriz  $S$  representa um documento  $d$ . Isto significa que no corpus,  $\mathbf{d}$  ainda é o documento de texto fonte original, mas dentro do nosso programa de computador, é representado pela coluna  $\mathbf{d}$  da Matriz  $S$ . Esta coluna é uma distribuição de números sobre o espaço de todas as palavras que existem no corpus. Representar cada documento no espaço de todas as palavras em todo o corpus é o maior valor desta matriz. Todos os documentos são representados nos mesmos termos. As células onde esses números são mais altos correspondem aos termos mais fortemente presentes no documento que também são informativos em todo o corpus. Porque os documentos são efetivamente representados como vetores, é muito fácil calcular o ângulo entre dois vetores

$$\cos(\theta) = \frac{\mathbf{d}_i \cdot \mathbf{d}_j}{\|\mathbf{d}_i\| \cdot \|\mathbf{d}_j\|}$$

Expand a esta fórmula abaixo e calcule o cosseno dos ângulos entre os possíveis pares de documentos no exemplo desta nota (dica: há três pares possíveis).



## 12.5 Notas Adicionais

Os principais conceitos/ideias que precisa entender e explicar são os seguintes:

1. O que queremos dizer com bag of words (saco de palavras)?
2. Por que o TF sozinho não é bom o suficiente para a representação matricial de um corpus?
3. O que são as linhas na representação matricial de um corpus?
4. O que são as colunas na representação matricial de um corpus?
5. O que conseguimos com a representação matricial de um corpus? Explique em detalhe.
6. Qual é a utilidade de calcular o ângulo entre dois documentos de uma determinada matriz  $S$ ?
7. O que significa quando dizemos que o IDF é um “modulador”?
8. O que queremos dizer com “lema” quando fazemos lematização das palavras?

O próximo tema focará o processamento da matriz  $S$  para encontrar tópicos usando fatorização matricial.