



Modelagem Probabilística de Tópicos

Oficina

Denio Duarte
duarte@uffs.edu.br



**UNIVERSIDADE
FEDERAL DA
FRONTEIRA SUL**
CAMPUS CHAPECÓ



Agenda

- Motivação
- Aprendizado de Máquina
- Modelagem de Tópicos
 - LDA (Latent Dirichlet Allocation)
- LDA Gensim

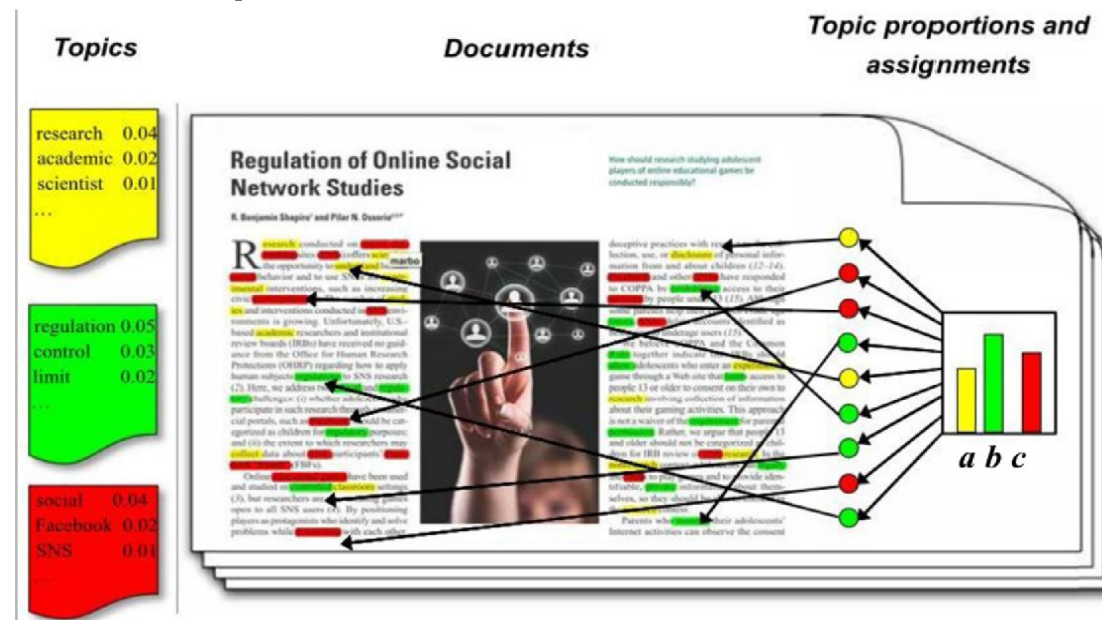
Motivação

- Documentos são produzidos todos os dias
- A Web é uma fonte “quase infinita” de documentos
 - Como classificá-los
 - Como consultá-los



Motivação

- Documentos podem conter classes:
 - Esporte, política, economia, entre outros
- As palavras dos documentos podem ser organizadas para definir tais classes



Fonte:

<https://www.semanticscholar.org/paper/Semantic-search-for-public-opinions-on-urban-A-Ma-Zhang/93f134aa1feccdcdfca2e61142311dc648fc54c1>

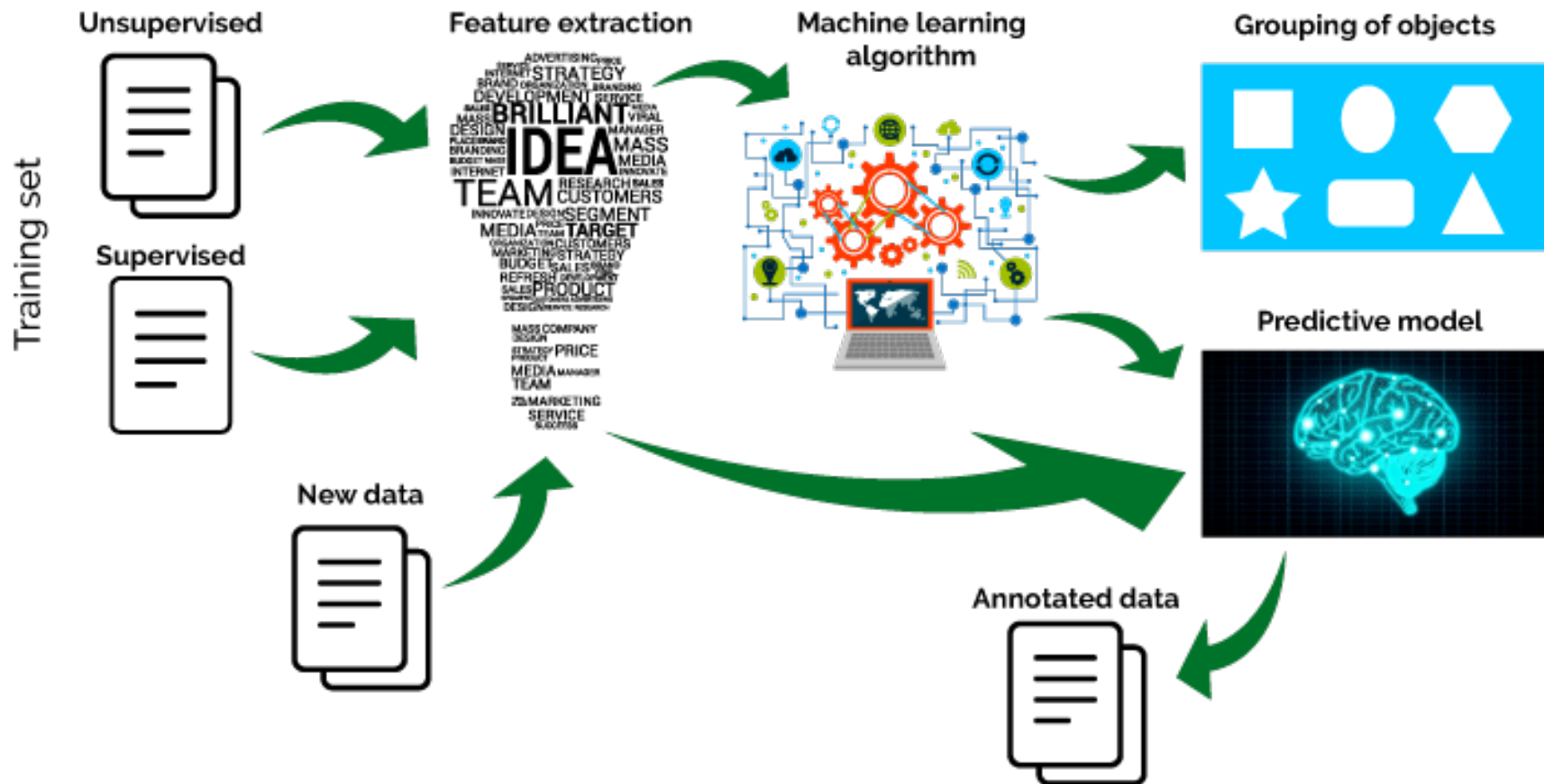


Motivação

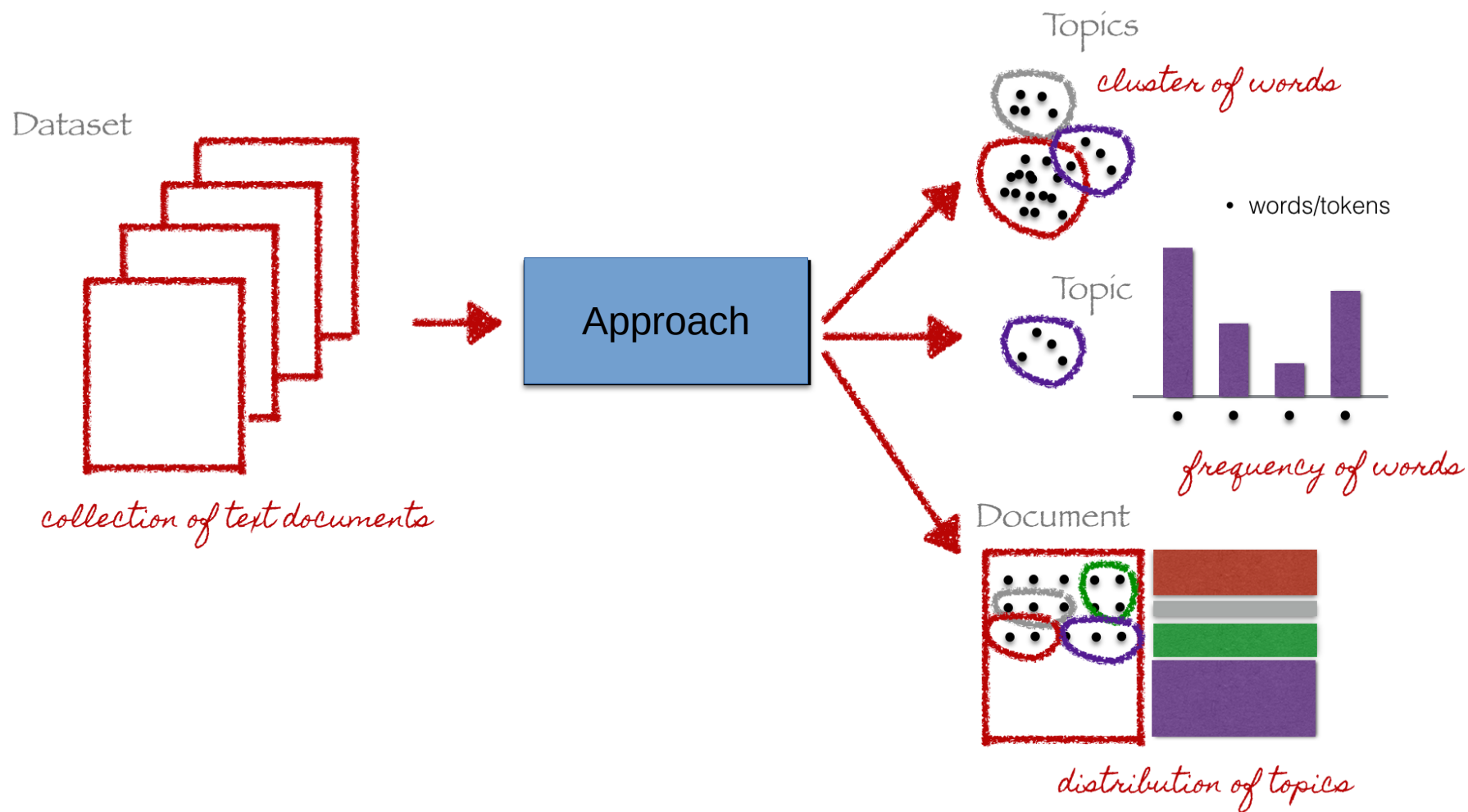
- Aprendizado de máquina é uma ferramenta que pode auxiliar nesta tarefa
- Abordagens de modelagem de tópicos estão sendo, cada vez mais, utilizadas nesta área de estudo

Aprendizado de Máquina

Machine Learning



Modelagem de Tópicos



LDA

- Documentos são visto como saco-de-palavras (bag-of-words)
 - A ordem das palavras não importa

DOCUMENTO A:

Ciência da computação é a ciência que estuda as técnicas, metodologias e instrumentos computacionais, que automatiza processos e desenvolve soluções baseadas no uso do processamento digital. Não se restringe apenas ao estudo dos algoritmos, suas aplicações e implementação na forma de software, extrapolando para todo e qualquer conhecimento pautado no computador, que envolve também a telecomunicação, o banco de dados e as aplicações tecnológicas que possibilitam atingir o tratamento de dados de entrada e saída, de forma que se transforme em informação. Assim, a Ciência da Computação também abrange as técnicas de modelagem de dados e os protocolos de comunicação, além de princípios que abrangem outras especializações da área.

BAG-OF-WORDS de A:

automatiza telecomunicação protocolos princípios especializações instrumentos banco processos transforme soluções computação ciência abrangem computador entrada pautado computacionais técnicas ciência algoritmos comunicação estuda desenvolve restringe uso não dados implementação tratamento metodologias forma possibilitam software processamento técnicas área abrange ciência computação extrapolando saída forma digital atingir baseadas nformação tecnológicas aplicações envolve aplicações estudo conhecimento dados modelagem dados

- Uso de técnicas de tokenization, stop-words, stemming e lemmatization



LDA

- Stop-words
 - Elimina as palavras “não” úteis e muito frequentes
e, ou, até, mais, mas, porém, não, sim, ...
- Tokenization
 - O texto é transformado em palavras:
Escola Regional de Banco de Dados → Escola, Regional, Banco, Dados

LDA

- Stemming
 - Heurística para cortar as palavras (geralmente sufixos):
studies → stud
studied → stud
studying → stud
- Lemmatization
 - Usa um vocabulário para padronizar as palavras:
am, is, are → be
studies → study
happiness, happy → happi

LDA

- A coleção de documentos de entrada é transformado em um corpo de palavras
 - *Corpus* (ou dicionário)

Doc 1: O restaurante universitário vai oferecer carnes de gado e de frango de qualidade.

Doc 2: No cardápio de segunda-feira será carne com legumes.

Doc 3: Legumes cozidos combinam com frango assado.

Doc 4: A limpeza do restaurante universitário é realizada com equipamentos automáticos.

Doc 5: Alguns equipamentos trabalham com detergentes para a limpeza do piso.



Doc 1: restaurante universitário ir oferece carne gado frango qualidade

Doc 2: cardápio segunda-feira carne legume

Doc 3: legume cozido combina frango assado

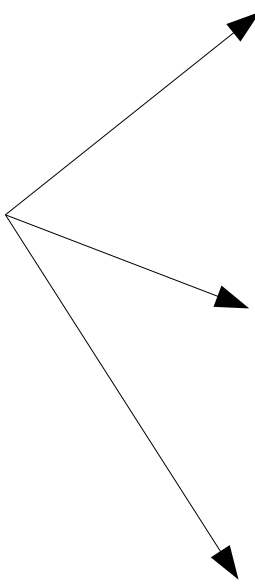
Doc 4: limpeza restaurante universitário realiza equipamento automático

Doc 5: equipamento trabalha detergente limpeza piso

LDA

- Construção do *corpus* (ou dicionário)

Doc 1: restaurante universitário ir oferece carne gado frango qualidade
Doc 2: cardápio segunda-feira carne legume
Doc 3: legume cozido combina frango assado
Doc 4: limpeza restaurante universitário realiza equipamento automático
Doc 5: equipamento trabalha detergente limpeza piso



Id	word
0	restaurante
1	universitário
2	ir
3	oferece
4	carne
5	gado
6	frango
7	qualidade
8	cardápio
9	segunda-feira
10	legume
11	cozido
12	combina
13	assado
14	limpeza
15	realiza
16	equipamento
17	automático
18	trabalha
19	detergente
20	piso

LDA

- Os documentos são transformados em id's e frequências

Id	word
0	restaurante
1	universitário
2	ir
3	oferece
4	carne
5	gado
6	frango
7	qualidade
8	cardápio
9	segunda-feira
10	legume
11	cozido
12	combina
13	assado
14	limpeza
15	realiza
16	equipamento
17	automático
18	trabalha
19	detergente
20	piso

Doc 1: restaurante universitário ir oferece carne gado frango qualidade
Doc 2: cardápio segunda-feira carne legume
Doc 3: legume cozido combina frango assado
Doc 4: limpeza restaurante universitário realiza equipamento automático
Doc 5: equipamento trabalha detergente limpeza piso

Doc 1: (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1) (7,1)
Doc 2: (8,1) (9,1) (4,1) (10,1)
Doc 3: (10,1) (11,1) (12,1) (6,1) (13,1)
Doc 4: (14,1) (0,1) (1,1) (15,1) (16,1) (17,1)
Doc 5: (16,1) (18,1) (19,1) (14,1) (20,1)

LDA

- Uma matrix (esparsa) $M_{doc\ id \times word\ id}$ é construída
 - Cada célula contém o número de ocorrência da palavra

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1

Id word
0 restaurante
1 universitário
2 ir
3 oferece
4 carne
5 gado
6 frango
7 qualidade
8 cardápio
9 segunda-feira
10 legume
11 cozido
12 combina
13 assado
14 limpeza
15 realiza
16 equipamento
17 automático
18 trabalha
19 detergente
20 piso

Doc 1: (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1) (7,1)
Doc 2: (8,1) (9,1) (4,1) (10,1)
Doc 3: (10,1) (11,1) (12,1) (6,1) (13,1)
Doc 4: (14,1) (0,1) (1,1) (15,1) (16,1) (17,1)
Doc 5: (16,1) (18,1) (19,1) (14,1) (20,1)

Doc 1: restaurante universitário ir oferece carne gado frango qualidade
Doc 2: cardápio segunda-feira carne legume
Doc 3: legume cozido combina frango assado
Doc 4: limpeza restaurante universitário realiza equipamento automático
Doc 5: equipamento trabalha detergente limpeza piso

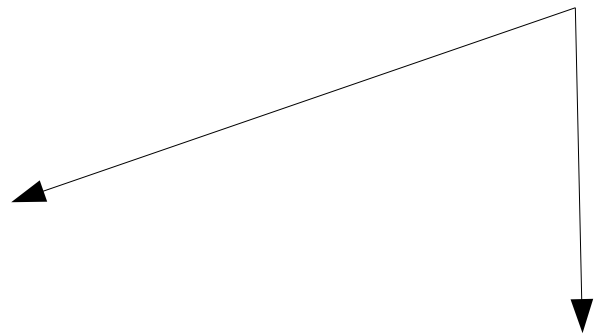
LDA

- $M_{doc\ id \times word\ id}$ é base para construção de duas outras matrizes
 - $D \times K$ e $K \times W$, onde K é o número de tópicos definidos

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1

$D \times K$ $k=3$

	0	1	2
1	P_{10}	P_{11}	$P_{12} = 1.0$
2	P_{20}	P_{21}	$P_{22} = 1.0$
3	P_{30}	P_{31}	$P_{32} = 1.0$
4	P_{40}	P_{41}	$P_{42} = 1.0$
5	P_{50}	P_{51}	$P_{52} = 1.0$



$K \times W$

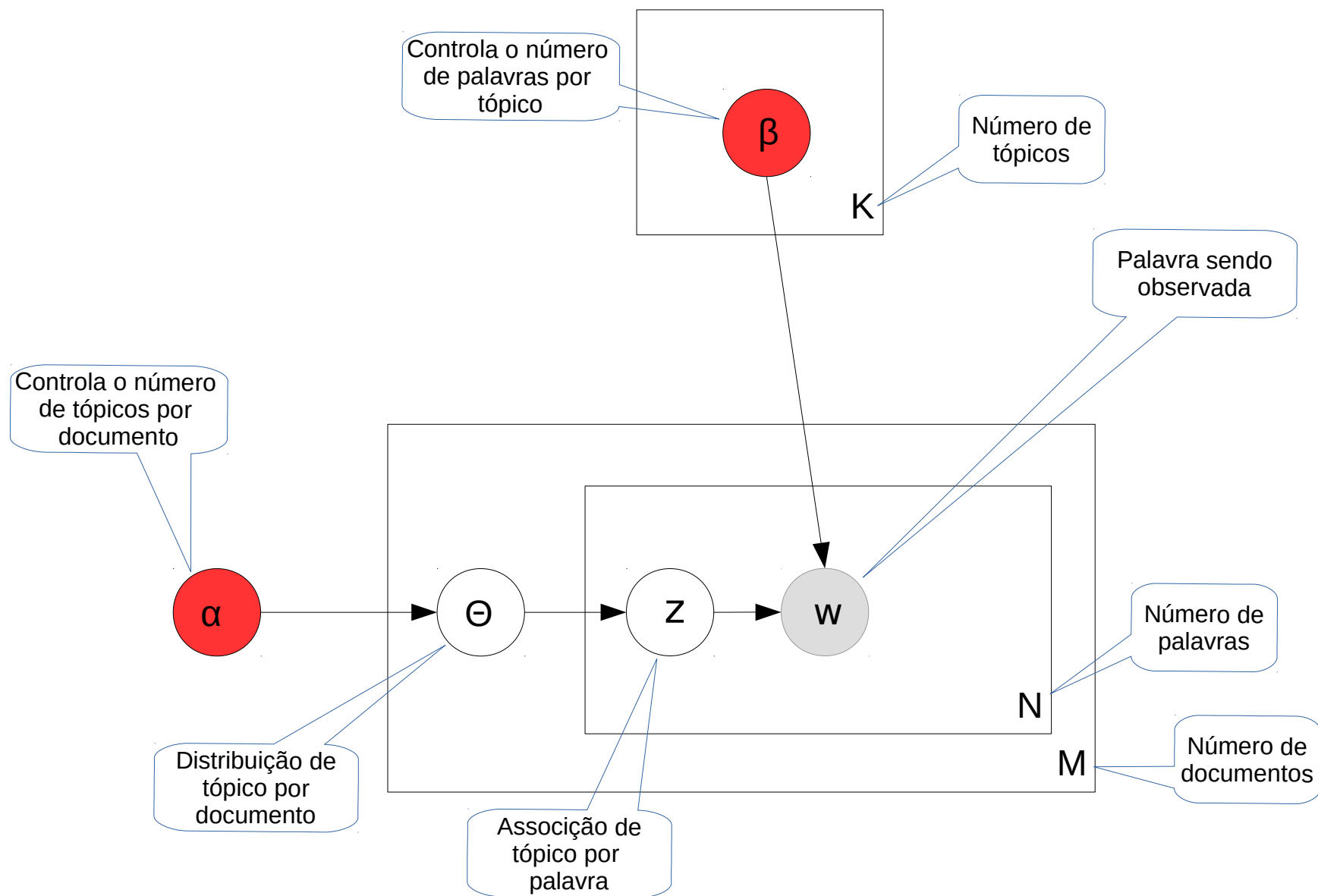
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	P_{00}	P_{01}	P_{02}	P_{03}	P_{04}	P_{05}	P_{06}	P_{07}	P_{08}	P_{09}	P_{010}	P_{011}	P_{012}	P_{013}	P_{014}	P_{015}	P_{016}	P_{017}	P_{018}	P_{019}	$P_{020} = 1.0$
1	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{110}	P_{111}	P_{112}	P_{113}	P_{114}	P_{115}	P_{116}	P_{117}	P_{118}	P_{119}	$P_{120} = 1.0$
2	P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}	P_{26}	P_{27}	P_{28}	P_{29}	P_{210}	P_{211}	P_{212}	P_{213}	P_{214}	P_{215}	P_{216}	P_{217}	P_{218}	P_{219}	$P_{220} = 1.0$



LDA

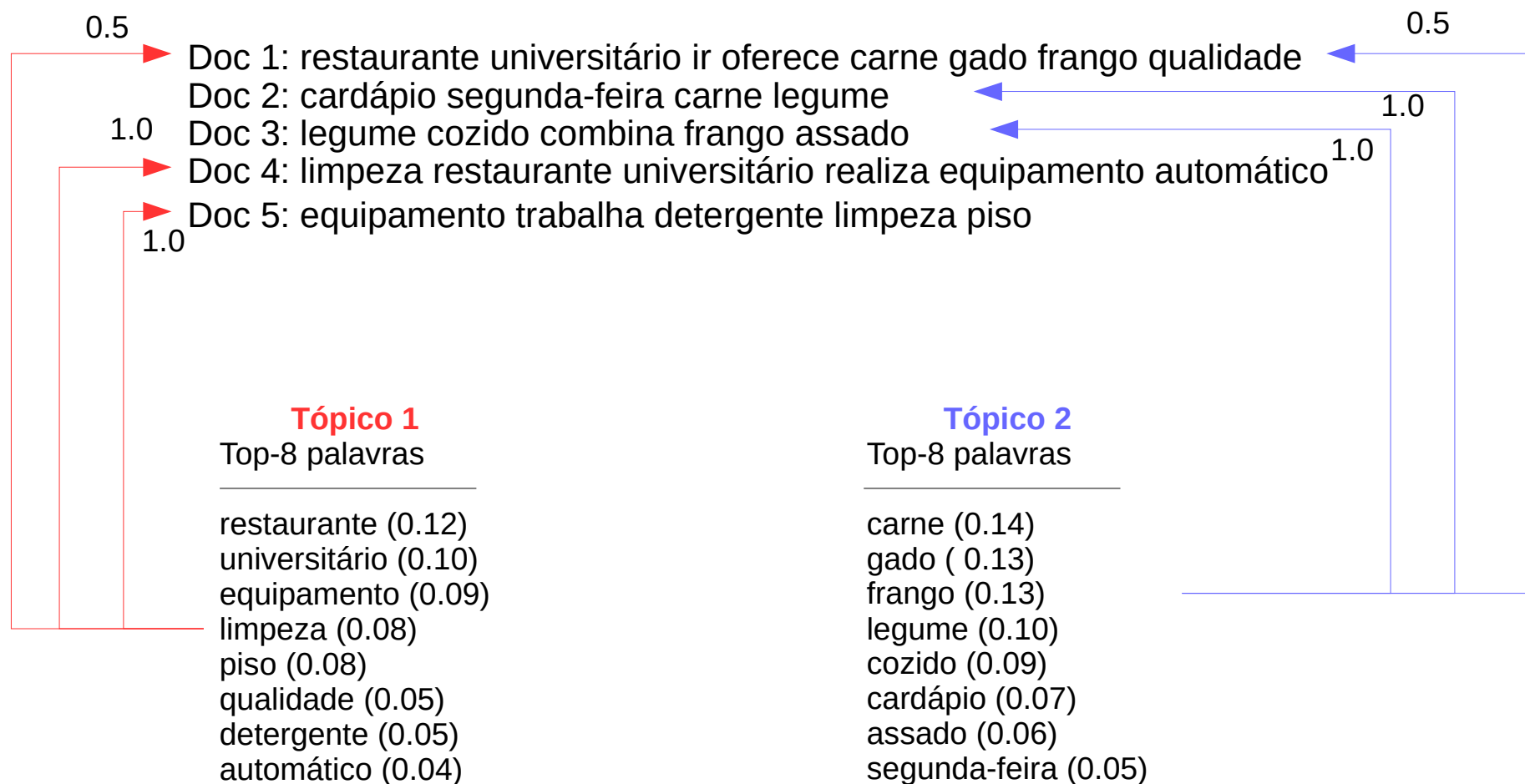
- Inicialmente, as palavras são associadas aos tópicos de forma randômica
 - A forma de distribuição segue a distribuição de Dirichlet
 - Latent Dirichlet Allocation
 - A alocação das probabilidades dos tópicos para cada palavra (distribuição desconhecida) é feita utilizando a distribuição de Dirichlet

LDA



LDA

- Distribuição dos tópicos por documento e palavras



Python

- Linguagem interpretada
- Os blocos de comandos são criados a partir de indentações

```
laço:  
    comando_1  
    :  
    comando_n  
fora do laço
```

Python

- Bibliotecas são importadas através do comando import

```
import numpy as np ## biblioteca para gerenciar matrizes
## métodos para eliminar stop words
from nltk.corpus import stopwords
## métodos para lemmatizar o document
from nltk.stem.wordnet import WordNetLemmatizer
## eliminar pontuações
import string
## biblioteca para utilizar os métodos LDA
import gensim
## métodos para construir o corpus dos documentos
from gensim import corpora
```

Python

- Pre-processamento (<https://goo.gl/8h8qj5>)

```
fdoc=open(collection_name) #abre arquivo (1 linha, 1 doc)
docs=fdoc.readlines() # carrega docs para a memória (lista de str)
## extrair os tokens (primeiro documento da coleção)
docs[0].split()
# Se docs[0]='Nobody is suppose to be here. So, get out!'
# Resultado ['Nobody','is','suppose','to','be','here.','So','get','out!']
# Reconstruir a string inicial
` `.join(['a','b','c']) ## Resultado 'a b c'
`-`.join(['a','b','c']) ## Resultado 'a-b-c'
## abordagem para acessar todos os tokens de um documento
[w for w in docs[0].split()]
## padronizar minúsculo
docs[0]=docs[0].lower()
# atualizar o primeiro documento sem as stop-words
stop = set(stopwords.words('english')) ## stop é um conjunto
## verifica palavra por palavra (token) e retorna aquelas que não
## estão no conjunto de stop-words
docs[0]=' ` `.join([w for w in docs[0].split() if w not in stop])
## 'nobody suppose here. so, get out!'
```

Python

Original: docs[0]='Nobody is suppose to be here. So, get out!'

- Pre-processamento (<https://goo.gl/8h8qj5>)

```
# removendo pontuações
pont=set(string.punctuation) #pont é plium conjunto
## percorre caracter por caracter e substitui a pontuação por nada
docs[0]=''.join(ch for ch in docs[0] if ch not in pont)
## 'nobody suppose here so get out'
## instancia um objeto lemmatizer
lemma = WordNetLemmatizer()
docs[0]=' '.join(lemma.lemmatize(w) for w in docs[0].split())
## continua igual 'nobody suppose here so get out'
## instancia um objeto stemmer utilizando a abordagem Porter
stpo=nltk.stem.PorterStemmer()
docs[0]=' '.join(stpo.stem(w) for w in docs[0].split())
## 'nobodi suppos here so get out'
```

Final: docs[0]='nobodi suppos here so get out'

Python

- Construindo o modelo de tópicos
 - Biblioteca gensim

```
# corpora auxilia na criação do dicionário (corpus)
from gensim import corpora
# gensim possui o método que implementa LDA
import gensim.models.ldamodel as gllda
## criar o corpus
dict = corpora.Dictionary(colecao_pre_tratada)
## filtrar algumas palavras do corpo
dict.filter_extremes(no_below=2, no_above=0.8, keep_n=500)
## as palavras que aparecem em apenas 2 documentos são descartadas
## as palavras que aparecem em mais de 80% dos documentos são descartadas
## o vocabulário conterá as primeiras 500 palavras mais frequentes
# transforma o documento em word id's
doc_ids=[dict.doc2bow(doc) for doc in colecao_pre_tratada]
## instancia e constroi o modelo, armazena em lda
lda = gllda.LdaModel(docs_id,num_topics=K,id2word=dict)
```

Python

- Navegando no modelo

```
## retornar as top n palavras do tópicos topic_id
w_top=lda.get_topic_term(topic_id,topn=n_palavras)
# w_top[0][0] word_id da primeira palavra
# w_top[0][1] probabilidade da primeira palavra
# a retorna a palavra codificada como word_id
dict.get(word_id)
#recuperar top-10 palavras dos tópicos
tpcs=[]
for t in range(K): ## K = número de tópicos
    words=lda.get_topic_term(t,topn=10)
    tpcs.append([dict.get(w[0]) for w in words])
## tpcs[0] tem as top-10 do tópico 0 (primeiro) e assim por diante
## retornar os tópicos de um documento (deve estar codificado)
## tópicos do primeiro documento (todos)
lda.get_document_topics(doc_ids[0],minimum_probability=0)
```


Python

- Avaliando o modelo

```
## biblioteca com as métricas
import gensim.models.coherencemodel as cm
## avalia todo o modelo utilizando a métrica u_mass
mycm=cm.CoherenceModel(model=lda, corpus=dict, coherence='u_mass')
mycm.get_coherence() ## retorna o valor da métrica para todos
## retorna o score para o primeiro documento
mycm=cm.CoherenceModel(topics=[tpcs[0]], dictionary=dict, texts=cole
cao_pre_tratada, coherence='u_mass')
## outras métricas c_uci, c_npmi e c_v
```