



**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

LEONARDO BIANCHINI

**ANÁLISE EXPLORATÓRIA DOS TÓPICOS NO STACK OVERFLOW
USANDO LDA (LATENT DIRICHLET ALLOCATION)**

**CHAPECÓ
2018**

LEONARDO BIANCHINI

**ANÁLISE EXPLORATÓRIA DOS TÓPICOS NO STACK OVERFLOW
USANDO LDA (LATENT DIRICHLET ALLOCATION)**

Trabalho de conclusão de curso de graduação
apresentado como requisito para obtenção do
grau de Bacharel em Ciência da Computação da
Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Denio Duarte

CHAPECÓ

2018

Bianchini, Leonardo

Análise exploratória dos tópicos no Stack Overflow usando LDA
(Latent Dirichlet Allocation) / por Leonardo Bianchini. – 2018.

38 f.: il.; 30 cm.

Orientador: Denio Duarte

Monografia (Graduação) - Universidade Federal da Fronteira Sul,
Ciência da Computação, Curso de Ciência da Computação, SC, 2018.

1. Modelagem de tópicos. 2. Stack Overflow. 3. LDA. 4. Tópico.
5. Análise exploratória. 6. Métricas. 7. Coerência. I. Duarte, Denio.
II. Título.

© 2018

Todos os direitos autorais reservados a Leonardo Bianchini. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: leonardobianchini7@gmail.com

LEONARDO BIANCHINI

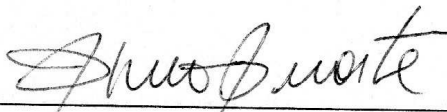
**ANÁLISE EXPLORATÓRIA DOS TÓPICOS NO STACK OVERFLOW
USANDO LDA (LATENT DIRICHLET ALLOCATION)**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Denio Duarte

Aprovado em: 03 \ 07 \ 2018

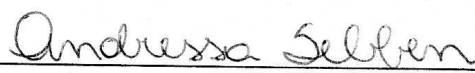
BANCA EXAMINADORA:



Dr. Denio Duarte - UFFS



Dr. Guilherme Dal Bianco - UFFS



Ma. Andressa Sebben - UFFS

RESUMO

A modelagem de tópicos é um problema de aprendizado de máquina, que visa extrair, dada uma coleção de documentos, os principais tópicos que representem os assuntos abordados pela coleção. Os documentos podem ser gerados a partir de diferentes distribuições sobre tópicos, sendo os tópicos formados por uma distribuição probabilística de palavras. Para inferir o conjunto de tópicos que geraram uma coleção de documentos, usam-se técnicas probabilísticas que fazem o processo reverso. Nesse trabalho, realiza-se uma análise exploratória na base de dados do *Stack Overflow*, e para tal, utiliza-se da modelagem de tópicos para a extração das informações desejadas, aplicando o LDA (*Latent Dirichlet Allocation*) para extrair os tópicos da base de dados. Como resultado, são obtidos os tópicos que representam a coleção, sendo mais recorrentes assuntos ligados à programação web, *mobile* e controle de versão. Além disso, são comparados os valores de tópicos, avaliados a partir de métricas que verificam a coerência entre suas palavras, identificando, dentre os valores analisados, o número de 50 tópicos com os melhores resultados para representar a coleção.

Palavras-chave: Modelagem de tópicos. Stack Overflow. LDA. Tópico. Análise exploratória. Métricas. Coerência.

ABSTRACT

Topic modeling is a machine learning problem, which aims to extract, given a collection of documents, the main topics that represent the subjects covered by the collection. Documents can be generated from different distributions on topics, the topics being formed by a probabilistic distribution of words. To infer the set of topics that generated a collection of documents, apply probabilistic techniques that make the process reverse. In this work, an exploratory analysis is performed in the Stack Overflow database, and for this purpose, it is used the topic modeling to extract the desired information, applying the Latent Dirichlet Allocation (LDA) to extract the topics from the database. As a result, the topics that represent the collection are obtained, with more recurring themes related to web programming, textit mobile, and version control. In addition, the values of topics are compared, evaluated from metrics that verify the coherence of their words, identifying, among the analyzed values, the number of 50 topics with the best results to represent the collection.

Keywords: Topic Modeling. Stack Overflow. LDA. Topic. Exploratory analysis. Metrics. Coherence.

LISTA DE FIGURAS

Figura 2.1 – Distribuição de tópicos em um documento (BLEI, 2012).	14
Figura 2.2 – Representação do modelo gráfico (FALEIROS, 2016; BLEI, 2012).	16
Figura 2.3 – Exemplo de <i>post</i> no <i>Stack Overflow</i> (Stack Overflow, 2017).	20

LISTA DE TABELAS

Tabela 2.1 – Exemplo da distribuição de ϕ_k .	18
Tabela 2.2 – Exemplo da distribuição de θ_j .	18
Tabela 4.1 – Características da base de dados.	27
Tabela 4.2 – Exemplo de <i>post</i> antes do processo de limpeza.	28
Tabela 4.3 – Exemplo de <i>post</i> depois do processo de limpeza.	28
Tabela 5.1 – Características da base de dados total.	31
Tabela 5.2 – Características da base de dados para o ano de 2017.	32
Tabela 5.3 – Top-5 tópicos de uma execução com $k = 50$.	32
Tabela 5.4 – Média <i>top-5</i> tópicos de cada k .	33
Tabela 5.5 – Média <i>top-10</i> tópicos de cada k .	33
Tabela 5.6 – Média <i>bottom-5</i> tópicos de cada k .	34
Tabela 5.7 – Média <i>bottom-10</i> tópicos de cada k .	34
Tabela 5.8 – Média geral de cada k para todos os tópicos.	34
Tabela 5.9 – Aplicação do T-Test	35

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Objetivos	10
1.1.1 Geral	10
1.1.2 Específicos	11
1.2 Justificativa	11
1.3 Estrutura do Trabalho	12
2 MODELAGEM PROBABILÍSTICA DE TÓPICOS	13
2.1 Tópicos	13
2.2 Modelos de Tópicos	14
2.3 Modelagem probabilística de tópicos	15
2.4 <i>Latent Dirichlet Allocation</i> (LDA)	16
2.5 <i>Gensim</i>	18
2.6 Base de dados: <i>Stack Overflow</i>	19
2.7 Métricas de avaliação	21
2.7.1 <i>Pointwise Mutual Information</i> (PMI)	21
2.7.2 <i>Normalized Pointwise Mutual Information</i> (NPMI)	22
2.7.3 <i>UCI Coherence</i>	22
2.7.4 <i>NPMI Coherence</i>	22
2.7.5 <i>UMass Coherence</i>	22
2.7.6 C_A <i>Coherence</i>	22
2.7.7 C_V <i>Coherence</i>	23
2.7.8 C_P <i>Coherence</i>	23
2.7.9 <i>Palmetto</i> - Ferramenta de Avaliação de Qualidade para Tópicos	24
3 TRABALHOS RELACIONADOS	25
4 PROJETO DE EXPERIMENTO	27
4.1 Configuração de ambiente	27
4.2 Base de dados	27
4.3 Pré-processamento dos dados	27
4.4 Aplicação do LDA	28
4.5 Pós-processamento dos dados	29
5 EXECUÇÃO E RESULTADOS	31
5.1 Execução	31
5.2 Resultados	32
5.3 Considerações finais	35
6 CONCLUSÃO	36
6.1 Trabalhos futuros	36
REFERÊNCIAS	37

1 INTRODUÇÃO

Modelagem de tópicos objetiva extrair, a partir de uma coleção de documentos, os principais tópicos, que representam os assuntos abordados pelos documentos da referida coleção.

Segundo (STEYVERS; GRIFFITHS, 2007), um modelo de tópico é um modelo generativo de documentos, sendo os documentos baseados na ideia de que são formados por uma mistura de tópicos. Os tópicos, por sua vez, são formados por uma distribuição probabilística de palavras. Os documentos com conteúdos diferentes podem ser gerados com distribuições diferentes sobre os tópicos. Para inferir o conjunto de tópicos que geraram uma coleção de documentos, são usadas técnicas estatísticas que fazem o processo inverso.

De modo geral, a modelagem de tópicos é uma tarefa que automatiza a extração de informações em grandes coleções de documentos, utilizando de métodos estatísticos para analisar as palavras do texto original e descobrir os tópicos, de maneira não-supervisionada.

No aprendizado de máquina, quando busca-se extrair informações a partir de grandes coleções de documentos, a modelagem de tópicos permite simplificar uma tarefa que seria onerosa caso executada manualmente. Essa tarefa deveria ser realizada através de uma análise minuciosa de todos os documentos da coleção. Assim, o problema da descoberta de tópicos está relacionado com o agrupamento de palavras que ocorrem frequentemente em documentos relacionados. De acordo com (BLEI, 2012), o problema computacional central para a modelagem de tópicos é o uso de documentos observados para inferir a estrutura de tópicos ocultos, que pode ser considerado como o processo reverso do modelo generativo de documentos. Assim, busca-se saber qual é a estrutura oculta que gerou a coleção observada.

Desse modo, a modelagem de tópicos é uma área bastante ampla e possui várias abordagens para a extração de tópicos. Este trabalho tem como foco o LDA (*Latent Dirichlet Allocation*) (BLEI; NG; JORDAN, 2003), para descobrir os tópicos na base de dados do *Stack Overflow*.

1.1 Objetivos

1.1.1 Geral

Desenvolver uma análise exploratória, a partir da extração dos principais tópicos abordados nas postagens pelos usuários do fórum *Stack Overflow*, utilizando como técnica de extração

o LDA.

1.1.2 Específicos

- Identificar qual assunto é representado dentro de cada tópico;
- Encontrar os hiper-parâmetros adequados para a obtenção dos tópicos;
- Identificar os tópicos mais recorrentes no *Stack Overflow*;
- Identificar um número de tópicos adequado que possa descrever coerentemente o período considerado no *Stack Overflow*; e
- Escolher métricas para verificar a qualidade e a coerência dos tópicos.

1.2 Justificativa

Sites de perguntas e respostas, também conhecidos como fóruns, geralmente dispõem de uma grande rede de usuários a fim de esclarecer dúvidas, compartilhar conhecimento e debater a respeito de algum assunto. Quando se trata de fórum sobre desenvolvimento, o *Stack Overflow* é um dos mais populares, sendo um dos principais fóruns de debate nessa área.

Mensalmente, milhares de postagens são feitas pela comunidade, gerando uma grande quantidade de dados, compreendendo uma variada gama de tópicos de interesse dos desenvolvedores. Esses dados fornecem um registro histórico do que é debatido pelos seus usuários. Esses registros, ao serem analisados e compreendidos, podem fornecer informações importantes acerca dos assuntos dentro do fórum, qual conteúdo está em alta - mais debatido (*hot topic*) e quais são os interesses dos desenvolvedores. Além disso, torna-se perceptível, através de uma análise temporal, o comportamento dos elementos supracitados.

De acordo com (BARUA; THOMAS; HASSAN, 2014), compreender esses tópicos pode ajudar os desenvolvedores compreender as tendências de uso, seja em linguagens ou plataformas, bem como permite aos vendedores comerciais avaliar a taxa de adoção de seus produtos. A compreensão é benéfica até ao próprio fórum, fazendo com que se perceba os padrões de uso e a forma pela qual os usuários interagem com a ferramenta.

Este trabalho, além de estudos sobre a modelagem probabilística de tópicos, visa a realização de uma análise exploratória, usando o LDA para extrair os tópicos a serem analisados. Os modelos probabilísticos de tópicos buscam descobrir estruturas temáticas ocultas em grandes

coleções de documentos. O LDA, por sua vez, é um modelo bayesiano completo embasado na geração de tópicos como distribuições de Dirichlet, descrevendo um modelo capaz de classificar documentos não conhecidos utilizando informações *a priori* (fornecidos previamente) (FALEIROS, 2016).

Para realizar a análise, visa-se preparar a base de dados do *Stack Overflow*, removendo palavras e fragmentos de texto que não venham a ser produtivos no problema em questão, para posteriormente fazer a extração dos tópicos.

Como resultado, serão obtidos os tópicos que representam a base de dados de acordo com as configurações propostas ao modelo. Além disso, será realizado um estudo para verificar a qualidade dos tópicos obtidos, utilizando métricas com o objetivo de avaliar os tópicos.

1.3 Estrutura do Trabalho

O restante deste trabalho está estruturado da seguinte forma: o próximo capítulo apresenta o referencial teórico. No Capítulo 3 são apresentados os trabalhos relacionados. No Capítulo 4 apresenta-se o projeto de experimento. Na sequência, o Capítulo 5 contém os resultados obtidos no desenvolvimento do trabalho. Por fim, no Capítulo 6 são apresentadas as conclusões.

2 MODELAGEM PROBABILÍSTICA DE TÓPICOS

Conhecimento e informação sempre foram transmitidos através das gerações, inicialmente com desenhos, evoluindo para os símbolos e posteriormente os alfabetos. Com o advento da tecnologia, mais meios passaram a ser utilizados a fim de armazenar conteúdo, como imagens, vídeos, etc. Ademais, a maneira mais simples de se guardar informação é na forma textual, de modo que existe uma grande quantidade de informação armazenada nesse formato. Com tamanha quantidade de informação textual existente, se torna humanamente impossível absorver ou captar informações relevantes desejadas. Logo, técnicas como a mineração de textos tem colaborado bastante para a obtenção de tais dados (FALEIROS, 2016).

A mineração de texto utiliza várias técnicas avançadas de mineração de dados, aprendizado de máquinas, recuperação de informação, extração de informação, linguística computacional e processamento de linguagem natural (FALEIROS, 2016). Logo, é relevante o desenvolvimento de maneiras automáticas para a extração de informações em textos, pois envolve a manipulação de dados não estruturados, sendo esta uma tarefa desafiadora.

Deste modo, pesquisadores de aprendizado de máquinas desenvolveram a modelagem probabilística de tópicos, que se trata de um conjunto de algoritmos com o objetivo de obter informações temáticas em grandes arquivos de texto. Segundo (BLEI, 2012), estes algoritmos são métodos estatísticos que analisam as palavras dos textos originais, visando descobrir os temas abordados, como eles se conectam e como mudam ao longo do tempo.

2.1 Tópicos

Os documentos e textos são formados por conjuntos de palavras que, em determinada ordem, atribuem sentido e representam um determinado assunto. Tais assuntos podem ser definidos como tópicos: conjuntos de palavras que ocorrem frequentemente em documentos que estão semanticamente relacionados, fazendo sentido dentro de determinado contexto.

Os tópicos podem ser obtidos a partir de uma técnica de pós processamento, realizada a partir das dimensões latentes (aleatórias) descobertas pela aplicação de modelos de tópicos.

De acordo com (BLEI, 2012), pode-se definir formalmente um tópico para ser uma distribuição em um vocabulário fixo. Por exemplo, um tópico de banco de dados tem palavras sobre banco de dados com alta probabilidade, assim como um tópico de inteligência artificial

contém palavras sobre inteligência artificial com alta probabilidade.

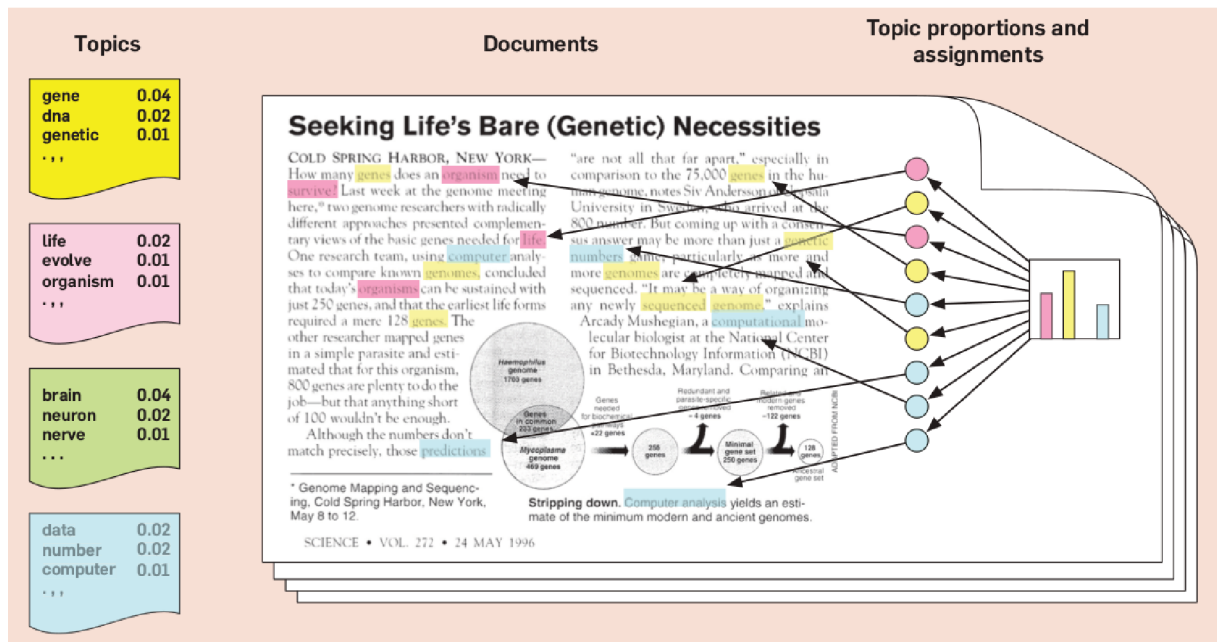


Figura 2.1 – Distribuição de tópicos em um documento (BLEI, 2012).

O exemplo apresentado na Figura 2.1 mostra o artigo "Seeking Life's Bare (Genetic) Necessities", que fala sobre o uso da análise de dados para determinar o número de genes que um organismo precisa para sobreviver. Fazendo um processo de classificação manual, foram sublinhadas palavras que se referem à um tema em comum com a mesma cor.

As palavras sublinhadas em azul referem-se a análise de dados, em amarelo sobre genética e em rosa a respeito de biologia evolutiva. De acordo com (BLEI, 2012), se destacadas todas palavras do artigo (excluindo palavras que não possuem conteúdo tópico - conhecidas como *stop words* - como "e", "mas", entre outras), será possível inferir que o artigo combina os tópicos de genética, biologia evolutiva e análise de dados em diferentes proporções.

A partir desses tópicos seria possível classificá-lo dentro de uma coleção de artigos científicos. A distribuição dos tópicos seguindo regras probabilísticas é dada pelos modelos de tópicos.

2.2 Modelos de Tópicos

Os modelos de tópicos procuram descobrir padrões latentes (*i.e.*, escondidos) para entender as relações entre documentos e palavras. Eles se baseiam na ideia de que os documentos são compostos por uma mistura de tópicos.

Como pode ser visto na Figura 2.1, modelos de tópicos são modelos generativos de documentos. Eles se apoiam em regras probabilísticas de amostragem, descrevendo como os índices de palavras podem ser gerados sob influência de variáveis latentes.

Como os documentos de uma coleção são formados por um mesmo conjunto de tópicos, o problema em descobrir quais tópicos geraram essa coleção se relaciona com o agrupamento de palavras que ocorrem em documentos relacionados. Os modelos de tópicos relacionam documentos e termos agrupando-os simultaneamente. Além disso, os documentos observados são utilizados para inferir a estrutura dos tópicos ocultos, fazendo o processo reverso do modelo generativo. Em outras palavras, os tópicos emergem da análise dos documentos a fim de representar a estrutura oculta que gerou tal coleção de documentos.

Para realizar o processo reverso da geração de documentos, iniciou-se essa linha de pesquisa, denominada por modelos probabilísticos de tópicos, que leva em consideração descobrir as variáveis latentes e as estruturas ocultas que geraram os documentos.

2.3 Modelagem probabilística de tópicos

A área de pesquisa em modelos probabilísticos de tópicos (*Probabilistic Topic Models*) surgiu em 2003, buscando satisfazer a necessidade em torno da obtenção de técnicas eficientes para a extração de informações em textos (FALEIROS, 2016). Os modelos probabilísticos de tópicos buscam descobrir estruturas temáticas ocultas em grandes coleções de documentos, que além de sua aplicação em documentos, podem ser usados em outros tipos de dados com atributos discretos. O modelo base da área é o LDA (*Latent Dirichlet Allocation*), que foi proposto por (BLEI; NG; JORDAN, 2003).

Os modelos probabilísticos de tópicos simplificam o processo de exploração de grandes volumes de dados na descoberta dos tópicos, que possuem valor semântico e formam grupos que frequentemente ocorrem juntos. Ao analisá-los, é possível inferir um tema ou assunto que ocorre num subconjunto de documentos.

O LDA é um modelo bayesiano completo e se baseia na geração dos tópicos como distribuições de *Dirichlet*, tendo a capacidade de classificar documentos não conhecidos e utilizar informações *a priori* (fornecidas previamente) (BLEI, 2012).

2.4 Latent Dirichlet Allocation (LDA)

O LDA (*Latent Dirichlet Allocation*) é um método de aprendizagem não supervisionado que busca encontrar tópicos em coleções de documentos. Conforme (BOLELLI; ERTEKIN; GILES, 2009), o LDA é um processo generativo que modela cada documento como uma mistura de tópicos, sendo cada tópico uma distribuição multinomial sobre palavras.

O LDA caracteriza-se por tomar os termos de cada documento como variáveis observáveis, enquanto as distribuições de tópicos são não observáveis. Os hiper-parâmetros, que são as distribuições de tópicos, são dados *a priori*.

A distribuição de Dirichlet, no processo generativo, é utilizada na distribuição de tópicos, sendo seu resultado utilizado para destinar as palavras aos documentos oriundas de diferentes tópicos. O processo generativo pode ser representado graficamente por meio de uma rede Bayesiana.

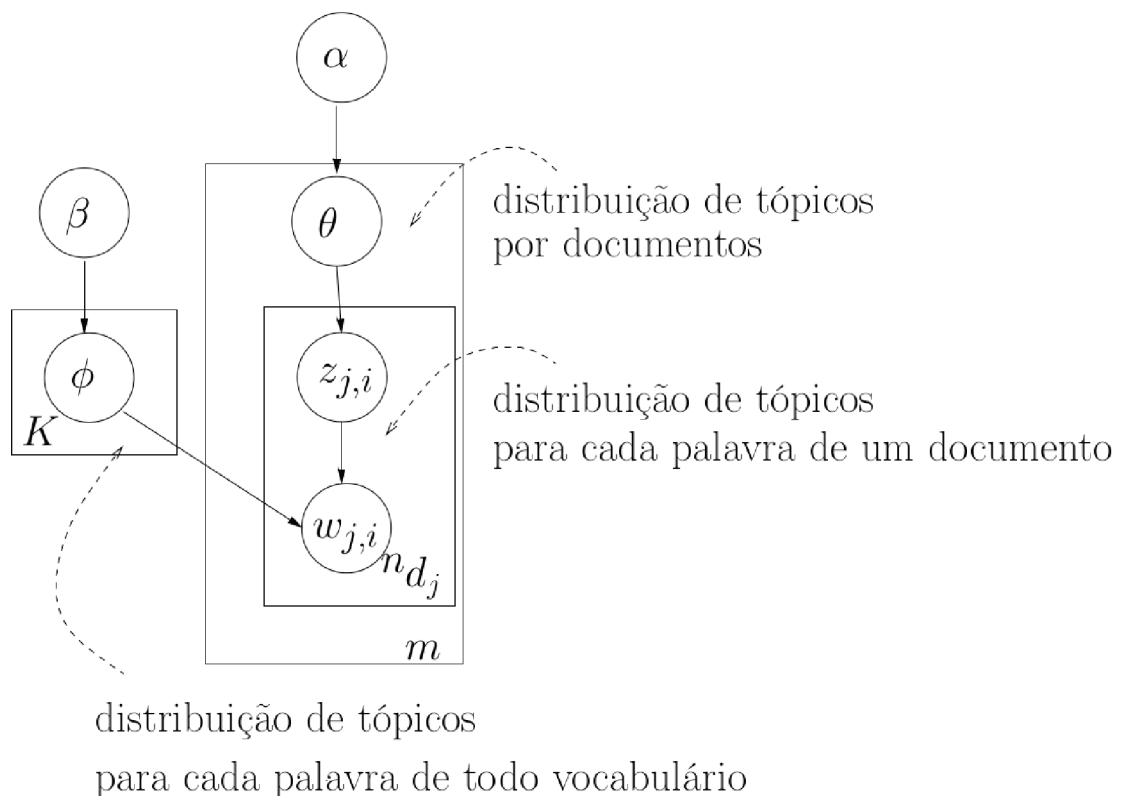


Figura 2.2 – Representação do modelo gráfico (FALEIROS, 2016; BLEI, 2012).

Onde, conforme a notação de (FALEIROS, 2016):

- K - número de tópicos.

- n - número de palavras do vocabulário.
- m - número de documentos.
- $n_{d,j}$ - número de palavras em um documento d_j , onde $1 \leq j \leq m$.
- θ - distribuição de tópicos por documentos.
- ϕ - distribuição dos tópicos sobre as palavras do vocabulário.
- θ_j - vetor com a proporção dos tópicos para o documento d_j , onde $1 \leq j \leq m$.
- ϕ_k - vetor com a proporção das palavras do vocabulário para o tópico k , onde $1 \leq k \leq K$.
- α - Priore da distribuição de Dirichlet, relacionada a distribuição documento-termo.
- β - Priore da distribuição de Dirichlet, relacionada a distribuição tópico-palavra.
- w_i - i -ésima palavra do vocabulário, onde $1 \leq i \leq n$.
- $w_{j,i}$ - palavra w_i observada no documento d_j , onde $1 \leq j \leq m$ e $1 \leq i \leq n$.
- $z_{j,i}$ distribuição de tópicos associado a palavra $w_{j,i}$ no documento d_j , onde $1 \leq j \leq m$ e $1 \leq i \leq n$.

Algoritmo 1: Pseudo-código processo gerador de um documento w

- 1 escolha $\theta \sim \text{Dirichlet}(\alpha)$.
 - 2 **para** cada uma das N palavras w_n **faça**
 - 3 | escolha uma palavra w_n de $p(w_n|\theta, \beta)$
 - 4 **fim**
-

Na Figura 2.2, cada retângulo representa um processo de repetição, sendo o número de vezes rotulado pela variável em seu interior. Segundo (FALEIROS, 2016), essa representação do modelo Bayesiano do LDA contém três níveis, sendo que o primeiro refere-se a distribuição de tópicos em todos os documentos, o segundo que distribui os tópicos para cada documento, e o terceiro que repete a distribuição dos tópicos internamente para as palavras de um documento. Esse último é o responsável por realizar a mistura dos tópicos.

Em nível de coleção, estão os hiper-parâmetros α e β , que determinam o comportamento dos tópicos. Quando o valor de α for alto, os documentos provavelmente compreenderão uma maior mistura de tópicos, e no caso contrário, a mistura será de poucos tópicos. Quando o valor

de β for alto, cada t3pico ter3 uma maior probabilidade de possuir misturas de v3rias palavras, e no caso contr3rio, o t3pico ser3 formado por poucas palavras.

A vari3vel ϕ_k 3 um vetor de probabilidade do vocabul3rio, sendo amostrada para cada t3pico k . Cada ϕ_k forma uma matriz $n \times K$, na qual as linhas representam palavras do vocabul3rio, e as colunas os t3picos, como pode ser visto na Tabela 2.1. Considerando $K = 4$, ou seja, quatro t3picos, e $n = 3$, ou seja, tr3s palavras. Perceba que a soma das probabilidades de uma palavra para cada t3pico 3 igual a um.

Tabela 2.1 – Exemplo da distribui33o de ϕ_k .

Palavra \ T3pico	k_1	k_2	k_3	k_4
w_1	0.02	0.28	0.55	0.15
w_2	0.00	0.33	0.51	0.16
w_3	0.49	0.11	0.17	0.23

A vari3vel θ_j est3 associada aos documentos, sendo θ uma matriz $m \times K$ no qual as colunas s3o os t3picos e as linhas os documentos, e θ_j a propor33o de t3picos para um documento d_j da cole33o, como pode ser observado na Tabela 2.2.

Tabela 2.2 – Exemplo da distribui33o de θ_j .

Documento \ T3pico	k_1	k_2	k_3	k_4
d_1	0.20	0.14	0.56	0.10
d_2	0.00	0.00	0.51	0.49
d_3	0.27	0.68	0.00	0.05

Por 3ltimo, a n3vel de palavras, est3o $z_{j,i}$ e $w_{j,i}$, amostradas para cada palavra w_i em cada documento d_j . J3 $z_{j,i}$ representa a atribui33o de um t3pico k para uma palavra w_i de documento d_j (FALEIROS, 2016).

2.5 Gensim

O *Gensim*¹ (ŘEHŮŘEK; SOJKA, 2010) 3 um kit de ferramentas *open source* implementado em *Python*, de modelagem de espa3o vetorial e modelagem de t3picos, que extrai automaticamente os t3picos com rela33o sem3ntica. Desenvolvido para lidar com grandes cole33es de documentos, descobre rela33es sem3nticas examinando padr3es de ocorr3ncia de palavras com, por exemplo, LSA (*Latent Semantic Analysis*) e LDA (*Latent Dirichlet Allocation*).

¹ radimrehurek.com/gensim/

2.6 Base de dados: *Stack Overflow*

O *Stack Overflow* é uma comunidade online onde usuários tiram dúvidas, aprendem e compartilham conhecimento sobre desenvolvimento. É uma rede social de perguntas e repostas, no qual os usuários buscam auxílio durante o processo de desenvolvimento ou manutenção de software. Segundo (TREUDE et al., 2012), o *Stack Overflow* conta com mais de 12 milhões de visitantes e 135 milhões de visualizações de páginas todo mês. De acordo com (ROCHA et al., 2016), todo o conteúdo das postagens fica disponível para a comunidade e assim, muitas vezes, várias das dúvidas já foram sanadas pelos seus usuários, sendo possível visualizar outras postagens já feitas.

No *Stack Overflow*, os usuários podem fazer perguntas e responder às questões existentes. Além disso, eles podem avaliar cada pergunta e comentário através de um sistema de votos, no qual atribuem pontos ao usuário que fez a intervenção. Esses pontos servem para "premiar" os usuários com distintivos ou emblemas de acordo com sua atividade no site. Ademais, o dono da pergunta pode avaliar as respostas, "aceitando" a resposta que lhe foi útil, que ficará destacada no *post*.


A Figura 2.3 é um exemplo de post no *Stack Overflow*, contendo título da pergunta, corpo do texto e *tags*. A resposta segue logo abaixo, contendo além do corpo do comentário, uma ferramenta de avaliação que é feita pelos próprios usuários. Ao lado esquerdo, seja da pergunta ou da resposta, há o recurso de votação.

Conforme (BARUA; THOMAS; HASSAN, 2014), o *Stack Overflow* dispõe seus dados publicamente no formato XML sob o *Creative Commons license*. Essa base de dados é composta pelos arquivos `badges.xml`, `comments.xml`, `posts.xml`, `users.xml` e `votes.xml`.


Para a análise em questão, o arquivo `posts.xml` fornece diversas informações. Na Figura 2.3 estão identificados os elementos presentes em tal arquivo, onde:

- (a) - título da postagem.
- (b) - corpo do texto.
- (c) - *tags*.
- (d) - votos.
- (e) - data e horário da postagem.

Topic models evaluation in Gensim (a)



Love remote work?
Find it on a new kind of career site



stackoverflow
JOBS

Get started

▲

2

▼

I've been experimenting with LDA topic modelling using [Gensim](#). I couldn't seem to find any topic model evaluation facility in Gensim, which could report on the perplexity of a topic model on held-out evaluation texts thus facilitates subsequent fine tuning of LDA parameters (e.g. number of topics). It would be greatly appreciated if anyone could shed some light on how I can perform topic model evaluation in Gensim. This question has also been posted on [metaoptimize](#).

(b)

(d) ★

1

lda gensim (c)

asked Oct 27 '13 at 8:11 (e)

Moses Xu (f)

1,037 • 14 • 27

share edit edited Oct 27 '13 at 9:07

add a comment

1 Answer

active

oldest

votes

(d)

▲

1

▼

Found the [answer](#) on the [gensim mailing list](#).

In short, the `bound()` method of `LdaModel` computes a lower bound on perplexity, based on a held-out corpus.

(b)

share edit

answered Nov 4 '13 at 5:03 (e)

Moses Xu (f)

1,037 • 14 • 27

✓ (g)

5 As of gensim 0.8.9, you can also use `model.log_perplexity(heldout)`, which is a convenience wrapper. – Radim May 31 '14 at 10:27

add a comment

Figura 2.3 – Exemplo de *post* no *Stack Overflow* (Stack Overflow, 2017).

- (f) - usuário responsável pela postagem.
- (g) - indicador da resposta aceita pelo dono da postagem.

Além dos itens supracitados, o arquivo `posts.xml` possui identificadores do tipo da postagem, se é pergunta ou resposta, contador de visualizações, identificador do *post*, entre outros.

2.7 Métricas de avaliação

As métricas são métodos utilizados para mensurar alguma coisa, de modo a trazer resultados importantes para a obtenção de dados, análise ou comparação de atributos. As métricas podem ser qualitativas (para avaliar se algo é bom ou não), ou quantitativas (que atribuem resultados numéricos ao que está sendo avaliado).

No contexto de modelagem de tópicos, as métricas podem colaborar na avaliação dos tópicos obtidos, permitindo determinar se os tópicos obtidos a partir dos algoritmos de modelagem de tópicos são realmente coerentes. Para tal, a avaliação poderia ser feita por humanos, entretanto, é uma tarefa onerosa (BLEI, 2012). Nesse contexto, o trabalho de (RÖDER; BOTH; HINNEBURG, 2015) implementa métricas já propostas por outros autores a fim de verificar a coerência com a base de dados utilizando ranqueamento por seres humanos.

Várias métricas foram propostas na literatura (veja em (RÖDER; BOTH; HINNEBURG, 2015) algumas delas). Neste trabalho, são utilizadas as seguintes métricas: PMI, NPMI, C_{UCI} , C_{NPMI} , C_{Umass} , C_A , C_V e C_P . Algumas métricas são baseadas em *sliding window* e *context window*:

- *sliding window* se trata de um subconjunto de palavras de tamanho N que "desliza" em qualquer direção sobre um conjunto maior. Por exemplo, considerando um conjunto $V = \{w_1, w_2, w_3, w_4, w_5, w_6\}$, uma *sliding window* de tamanho 3 pode conter $\{w_2, w_3, w_4\}$, deslizando, ela pode passar a ter $\{w_3, w_4, w_5\}$;
- *context window*, por sua vez, é um subconjunto com N palavras que antecedem ou sucedem uma determinada palavra. Considerando o mesmo conjunto, uma *context window* de tamanho 2 sobre w_4 corresponde a $\{w_2, w_3, w_4, w_5, w_6\}$;

2.7.1 Pointwise Mutual Information (PMI)

Pointwise Mutual Information (PMI) é um método utilizado para mensurar a associatividade entre duas palavras. Ela considera $P(w_i, w_j)$ a probabilidade de duas palavras w_i e w_j ocorrerem na mesma janela de palavras, $P(w_i)$ e $P(w_j)$ as probabilidades de w_i e w_j ocorrerem individualmente. A constante ϵ , por sua vez, serve para evitar o logaritmo de zero. A fórmula é dada a seguir:

$$PMI(w_i, w_j) = \log \left(\frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)} \right)$$

2.7.2 Normalized Pointwise Mutual Information (NPMI)

O *NPMI* (*Normalized Pointwise Mutual Information*), por sua vez, é uma normalização no *PMI*, a fim de se obter resultados no intervalo $[-1, 1]$, no qual 1 indica completa co-ocorrência entre as palavras, -1 nenhuma co-ocorrência e 0 significa independência entre as palavras.

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log(P(w_i, w_j))}$$

2.7.3 UCI Coherence

A *UCI Coherence* usa o *PMI* para calcular a coerência sobre todos os pares das *N-top words* de um tópico usando uma *sliding window* de tamanho 10.

$$C_{UCI} = \frac{2}{N \cdot (N - 1)} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^N PMI(w_i, w_j)$$

2.7.4 NPMI Coherence

Funciona de maneira semelhante a *UCI*, com a diferença de usar *NPMI* no lugar do *PMI*.

$$C_{NPMI} = \frac{2}{N \cdot (N - 1)} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^N NPMI(w_i, w_j)$$

2.7.5 UMass Coherence

Considera a relação entre a probabilidade das palavras dentro de um tópico com a sua ocorrência no cálculo da coerência. Assim, se um documento contém uma palavra de ordem maior dentro do tópico, a probabilidade de uma determinada palavra ocorrer será maior. Para cada palavra, ela calcula o logaritmo da probabilidade condicional para as palavras que a precedem no tópico, e também usa uma constante ϵ para evitar o logaritmo de zero.

$$C_{Umass} = \frac{2}{N \cdot (N - 1)} \cdot \sum_{i=2}^N \sum_{j=1}^{i-1} \log \left(\frac{P(w_i, w_j) + \epsilon}{P(w_j)} \right)$$

2.7.6 C_A Coherence

Utiliza de uma variação do *NPMI* com *context window* de tamanho 5 para mensurar a relação de todos os pares de palavras de um tópico. Além disso, ela utiliza uma variável γ para

atribuir peso maior para as maiores associatividades, e também usa ϵ para evitar o logaritmo de zero.

$$v_{ij} = NPMI(w_i, w_j)^\gamma = \left(\frac{PMI(w_i, w_j)}{-\log(P(w_i, w_j) + \epsilon)} \right)^\gamma$$

Após aplicação da fórmula acima, guardam-se os resultados num vetor para cada palavra do tópico, contendo as comparações com as demais.

$$\vec{v}_i = \{NPMI(w_i, w_i)^\gamma, NPMI(w_i, w_j)^\gamma, \dots, NPMI(w_i, w_j)^\gamma\}$$

Depois, é feita uma média aritmética sobre o cálculo da similaridade para o cosseno dos vetores de cada par de palavras distintas, assim obtendo-se o resultado.

$$C_A = \frac{1}{n} \cdot (\cos(\vec{v}_i, \vec{v}_j) + \dots + \cos(\vec{v}_m, \vec{v}_n))$$

Sendo n o número de pares de palavras distintas.

2.7.7 C_V Coherence

Funciona de maneira semelhante a C_A Coherence, entretanto, usa *sliding window* de tamanho 110. Além disso, ela difere da anterior na hora de calcular a similaridade dos cossenos: em vez de comparar os pares de vetores das palavras, ela compara o vetor de cada palavra com o vetor resultante das somas dos vetores de todas palavras.

$$\vec{v}_c = \sum_{i=1}^N \vec{v}_i$$

$$C_V = \frac{1}{N} \cdot \sum_{i=1}^N \cos(\vec{v}_i, \vec{v}_c)$$

2.7.8 C_P Coherence

A C_P Coherence é uma métrica que usa uma *sliding window* de tamanho 70, baseando-se na coerência de Fitelson, de acordo com a fórmula abaixo:

$$C_P = \frac{2}{N \cdot (N - 1)} \cdot \sum_{i=2}^N \sum_{j=1}^{i-1} m_f(w_i, w_j)$$

Onde calcula-se, para cada palavra do tópico, a m_f com as palavras que a precedem no tópico. A função denotada m_f calcula o grau de relação entre a palavra w_i com o conjunto $S(i)$. Nela (FITELSON, 2003) avalia uma palavra w_i no contexto formado por todos os subconjuntos

construídos a partir das demais palavras do tópico. $S(i)$ representa o conjunto de todos os subconjuntos formados sem a palavra w_i .

$$m_f(w_i, S(i)_j) = \frac{P(W_i|S(i)_j) - P(W_i|\neg S(i)_j)}{P(W_i|S(i)_j) + P(W_i|\neg S(i)_j)}$$

2.7.9 *Palmetto* - Ferramenta de Avaliação de Qualidade para Tópicos

O *Palmetto*² é uma ferramenta proposta por (RÖDER; BOTH; HINNEBURG, 2015) que visa mensurar a qualidade de tópicos. Ele dispõe de métricas que avaliam a coerência dos tópicos de acordo com a co-ocorrência das palavras no *Wikipedia*. Ela está disponível em uma versão *demo*³ que pode ser utilizada *online*, que limita a avaliação a 10 palavras, e uma versão mais completa de código aberto, disponível no *Github*⁴.

² <http://aksw.org/Projects/Palmetto.html>

³ <http://palmetto.aksw.org/palmetto-webapp>

⁴ <https://github.com/dice-group/Palmetto>

3 TRABALHOS RELACIONADOS

Existem vários trabalhos na literatura que utilizam LDA para a extração de tópicos em coleções de documentos. Blei (BLEI, 2012) apresenta uma revisão da literatura com os trabalhos mais relevantes. Porém, este trabalho foca na análise do fórum *Stack Overflow* utilizando LDA. Assim, foram selecionados dois trabalhos similares a esta proposta.

O trabalho de (BARUA; THOMAS; HASSAN, 2014) propõe uma metodologia para analisar o conteúdo textual das discussões no *Stack Overflow*, a fim de ajudar a comunidade da Engenharia de Software a entender as necessidades dos desenvolvedores. Como objetivos específicos, ele visa descobrir os principais tópicos de discussão, suas dependências subjacentes e tendências ao longo do tempo. Para tal, ele usa o LDA na extração dos tópicos da base de dados. A base de dados utilizada nesse trabalho contempla um período de 27 meses, de julho de 2008 a setembro de 2010. Além disso, contava com 3.474.987 *posts*, sendo 973.267 (28%) questões e 2.501.720 (72%) respostas.

Para descobrir os tópicos da base de dados, foi necessário fazer a extração dos dados em formato *xml* e pré-processamento de dados, no qual descarta-se o conteúdo não produtivo para a análise, como trechos de código, *stop words* e *tags*.

Para descobrir os tópicos, foi utilizada uma implementação do LDA. Como resultado, foi obtido um conjunto de temas, com os principais tópicos da base de dados, além de um conjunto de vetores com a adesão por tópico, representando o percentual de palavras no *post* que veio de cada documento.

Com isso, (BARUA; THOMAS; HASSAN, 2014) puderam observar que as perguntas de alguns tópicos levam a discussões em outros tópicos, e os tópicos que se tornam mais populares ao longo do tempo são o de desenvolvimento web, aplicações *mobile*, Git e MySQL.

Em (LINARES-VÁSQUEZ; DIT; POSHYVANYK, 2013), é apresentada uma análise exploratória sobre desenvolvimento *mobile*, também examinando a base de dados do *Stack Overflow*, a fim de obter dados mais específicos acerca do desenvolvimento móvel.

O que difere este trabalho do apresentado por (BARUA; THOMAS; HASSAN, 2014) é o tipo de informação analisada. Em (LINARES-VÁSQUEZ; DIT; POSHYVANYK, 2013), utiliza-se as *tags* para filtrar as perguntas, neste caso, relacionados ao desenvolvimento *mobile*, como *android*, *bada*, *blackberry*, *iphone*, *ios*, *java-me*, *phonegap*, *symbian*, *tizen*, *webos* e *windows-phone*.

Além disso, leva-se em consideração, neste trabalho, a questão de perguntas respondidas e não respondidas, e também de respostas aceitas. Quando usuários fazem perguntas no *Stack Overflow*, existe a possibilidade de que não sejam respondidas. Quando há respostas, elas podem ser avaliadas pelo dono da pergunta, que aceita quando julga que a resposta foi útil.

Com isso, compara-se dados específicos entre as duas maiores plataformas (*android e ios*), no qual conclui-se que a diferença entre perguntas não respondidas no *ios* é maior do que no *android*. Ademais, verificou-se também que há um subconjunto de tópicos mais propensos a terem respostas aceitas, como questões relacionadas aos tipos de dados, compatibilidade e *layout*.

Com relação às respostas aceitas, o autor conclui que a maioria das respostas aceitas são oriundas de usuários que trabalham com apenas uma tecnologia *mobile*. No entanto, também existem contribuidores que forneceram respostas aceitas relacionadas para mais de uma tecnologia.

Diferentemente das obras citadas anteriormente, o presente trabalho realiza a exploração de uma base de dados que contempla um período cronológico diferente, buscando apontar, por exemplo, os tópicos mais recorrentes e os melhores valores de tópicos, apoiados em métricas, para representar a coleção observada.

4 PROJETO DE EXPERIMENTO

Descreve-se neste capítulo como o experimento foi realizado, apresentando dados sobre o ambiente de desenvolvimento do trabalho, descrição da base de dados e etapas necessárias para reprodução do experimento.

4.1 Configuração de ambiente

Durante o desenvolvimento do projeto, utilizou-se de um computador *Lenovo Think-Centre M92p*, contando com processador *Intel Core i5-3470 CPU @ 3.20GHz x 4*, 16GB de memória RAM, placa gráfica *GeForce GT 610/PCIe/SSE2*, disco rígido com 1TB de capacidade, e sistema operacional *Ubuntu 16.04.4 LTS 64-bit*.

4.2 Base de dados

A base de dados utilizada neste projeto é a do *Stack Overflow*, disponibilizada através do *Wayback Machine*⁵, um arquivo denominado `Posts.xml`, contendo 12.1GB compactado e 60.5GB descompactado. Ela compreende todos os *posts* dispostos no período entre agosto de 2008 a março de 2018, e contém 38.485.046 documentos. A Tabela 4.1 detalha características da base de dados, com número de perguntas e repostas, quantidade total de palavras e de palavras únicas.

Tabela 4.1 – Características da base de dados.

Número de Perguntas	Número de Respostas	Quantidade de palavras	Quantidade de palavras únicas
14.995.834	23.489.212	772.757.313	2.001.814

4.3 Pré-processamento dos dados

Para poder trabalhar com os dados, faz-se necessário um pré-processamento do arquivo `Posts.xml`, executando-se um processo de preparação dos dados. Para o desenvolvimento desta tarefa, é necessário elaborar um algoritmo que, iterando nos *posts*, seja capaz de extrair o corpo do texto, identificado pela *tag Body*. Os demais campos presentes no *xml* não foram con-

⁵ archive.org/download/stackexchange/

siderados para o desenvolvimento deste trabalho. Dentro de cada texto aplicam-se os seguintes processos para a limpeza dos dados:

- Remoção de trechos de código: os trechos identificados no *xml* entre *tags* "`<code>`", foram descartados devido aos trechos de código serem similares a várias linguagens de programação;
- Remoção de pontuação: os sinais foram descartados para melhorar a classificação das palavras nos demais passos;
- Lematização: processo utilizado para deflexionar palavras, reduzindo-as ao seu lema;
- Stemmização: procedimento que reduz palavras flexionadas ao seu radical;
- Remoção de *stop words*: remove palavras de interrupção comuns à linguagem, que não possuem conteúdo tópico e não são interessantes ao modelo;
- Remoção de palavras com tamanho menor que 3: remove palavras que podem conter siglas e que não possuem conteúdo para a análise;

Após o procedimento de limpeza, cada *post* é registrado em um arquivo de saída em formato de texto. A Tabela 4.2 mostra como um *post* está registrado na base de dados, sem qualquer tratamento, enquanto a Tabela 4.3 mostra o mesmo *post* após a limpeza e *tokenizado*.

Tabela 4.2 – Exemplo de *post* antes do processo de limpeza.

```
<p>My problem was that <code>cable.js</code>was not included in <code>application.js</code> because I have my javascript files included explicitly by name and forgot to add cable.</p>
```

```
<p>Don't know if this Q&A will help anyone else, but if it does....</p>
```

Tabela 4.3 – Exemplo de *post* depois do processo de limpeza.

```
['problem','includ','javascript','file','includ','explicitli','forgot','cable','know','help']
```

4.4 Aplicação do LDA

Com o arquivo pré-processado (limpo), parte-se para a preparação dos dados para posterior aplicação do modelo para extração dos tópicos. Para tal, faz-se uso da biblioteca *Gensim*⁶

⁶ radimrehurek.com/gensim/

(ŘEHŮŘEK; SOJKA, 2010), gerando dois arquivos: dicionário e o *corpus*. O dicionário possui o registro das palavras contidas na base de dados, atribuindo um identificador único, e também um contador para as ocorrências de tal palavra na base de dados. O *corpus*, por sua vez, é uma combinação de todos os documentos de texto, sendo composto por uma representação matricial entre documentos e termos. O *corpus* é usado pelo LDA para procurar padrões na matriz documento-termo.

No processo de construção do dicionário e *corpus*, aplica-se um filtro para que eles não contenham palavras que estejam presentes em muitos ou poucos documentos, pois eles podem distorcer os resultados. Assim, define-se o limite inferior em 0.2% e o superior em 80%. Deste modo, os tópicos serão formados somente por palavras com ocorrências neste intervalo.

Dados dicionário e *corpus*, definiram-se os números de tópicos, denotados por k , em 50, 100, 150 e 200. Não existe um valor de k que satisfaça todas as situações, por isso buscaram-se alguns valores com base em um conhecimento prévio acerca da base de dados. O valor de k é determinante na busca por uma melhor representação da base de dados, definindo se os tópicos serão mais gerais ou detalhados. Além disso, utiliza-se os hiper-parâmetros α e β *default* do modelo, dados pela equação:

$$\alpha = \frac{1}{num_topics}$$

Feito isso, aplica-se o LDA através do *Gensim*, passando dicionário e *corpus* como parâmetro, executando para cada valor de k proposto. Para cada execução, guardam-se os resultados em arquivo no formato *model* para posterior avaliação dos tópicos extraídos.

Algoritmo 2: Script de execução do LDA

```

1 import gensim
2 from gensim import corpora
3 dictionary = gensim.corpora.Dictionary.load('dictionary.dict')
4 corpus = gensim.corpora.MmCorpus('corpus.mm')
5 Lda = gensim.models.ldamodel.LdaModel
6 ldamodel = Lda(corpus, num_topics=50, id2word = dictionary)
7 print(ldamodel.print_topics(num_topics=50, num_words=10))
8 ldamodel.save('lda/lda50.model')
```

4.5 Pós-processamento dos dados

Depois de obter-se os tópicos, é primordial realizar um processo de ajuste das palavras, isso devido aos processos realizados durante a limpeza da base de dados, principalmente a le-

matização e stemmização, que reduzem as palavras ao seu radical. Para tal, é necessário fazer o processo reverso, buscando a partir do radical reconstruir as palavras. Assim, manualmente, percorre-se todas as palavras dos tópicos, fazendo o ajuste necessário para normalizar as palavras.

A seguir, submetem-se os tópicos obtidos e normalizados para as métricas de avaliação. Para tal, aplica-se em cada tópico as métricas dispostas no *Palmetto Online Demo*⁷, armazenando os resultados em um arquivo *csv*.

Para avaliação, analisam-se os resultados das métricas para cada valor de tópicos, comparando as médias obtidas para cada valor de k .

Para validar as comparações entre os valores de k , em cada métrica, utiliza-se o teste *Student's T-Test (t-test)*. Com o teste, obtêm-se resultados no intervalo $[0,1]$, mensurando a confiança de uma afirmação. As hipóteses a serem verificadas podem ser vistas na Equação 4.1. Na hipótese H_1 , pode se afirmar que os valores de k são proporcionais caso o resultado do *t-test* seja $\alpha > 0,1$. Caso contrário, pode se afirmar que um valor k é melhor que o outro.

$$\begin{cases} H_1 : k X_1 = k X_2, & \text{se } \alpha > 0,1 \\ H_2 : k X_1 \neq k X_2, & \text{se } \alpha < 0,1 \end{cases} \quad (4.1)$$

Assim, com α definido em $0,1$, valida-se a hipótese H_2 , permitindo afirmar que X_1 obteve um resultado médio diferente de $k X_2$ com 90% de confiança.

O próximo capítulo apresenta os resultados dos experimentos.

⁷ palmetto.aksw.org/palmetto-webapp/

5 EXECUÇÃO E RESULTADOS

Neste capítulo é apresentado como foi realizado o processo de execução dos experimentos, seguindo o projeto apresentado no capítulo anterior, e os resultados a partir de sua execução, além de considerações finais sobre o trabalho.

5.1 Execução

Com base no projeto de experimento, foi obtida a base de dados necessária para a realização do trabalho. A seguir, realizou-se o pré-processamento dos dados, com o desenvolvimento de um algoritmo na linguagem *Python*, iterando nos posts a fim de extrair o corpo do texto e aplicar as técnicas de limpeza. Com esse processo, gerou-se um arquivo de texto contendo 7GB de dados.

A partir do texto limpo, partiu-se para a aplicação do LDA. Para tal, com a utilização do *Gensim*, percorreram-se os dados para a construção dos arquivos de dicionário e *corpus*. Ao término do processo, obtiveram-se os arquivos de acordo com a Tabela 5.1:

Tabela 5.1 – Características da base de dados total.

Número de Perguntas	Número de Respostas	Quantidade de palavras	Quantidade de palavras únicas
14.995.834	23.489.212	772.757.313	2.001.814

Feito isso, aplicou-se o LDA implementado pelo *Gensim*, passando dicionário e corpus como parâmetro, buscando-se obter inicialmente 50 tópicos. Entretanto, após 4 dias de execução ainda não haviam sido obtidos resultados, motivo pelo qual o processo foi abortado. Com isso, o processo foi reavaliado, a fim de contemplar uma base de dados com tamanho aceitável, mas que pudesse ser executada em tempo hábil. Definiu-se, então, analisar os dados referentes ao ano de 2017.

Para adequar o projeto, voltou-se ao processo de limpeza, aplicando um filtro para selecionar apenas os *posts* que estivessem no período especificado. Para tal, identificava-se através da tag "*creationDate*" a data de criação do documento, e realizava-se a limpeza e armazenamento dos dados que correspondessem às restrições impostas. Com isso, foram obtidos 5.113.521 documentos. A Tabela 5.2 apresenta dados referentes à base de dados, com número de perguntas e repostas, quantidade total de palavras e de palavras únicas.

Na sequência, foram refeitos também o dicionário e o *corpus*. Para a construção de tais

Tabela 5.2 – Características da base de dados para o ano de 2017.

Número de Perguntas	Número de Respostas	Quantidade de palavras	Quantidade de palavras únicas
2.331.406	2.782.115	103.705.956	1.953.725

arquivos, aplicou-se o filtro dos extremos, mantendo as palavras presentes em pelo menos 10 mil documentos, e descartando as que estão presentes em mais de 90%. O novo dicionário contou com 1314 palavras, enquanto o corpus foi constituído a partir dos 5.113.521 documentos, com 86.399.511 entradas diferentes de zero, em um arquivo de 1.2GB. Em seguida, aplicou-se o LDA para 50, 100, 150 e 200 tópicos, executando-se 5 vezes para cada número.

Com os tópicos extraídos, foi necessário normalizar as palavras antes de aplicá-las às métricas. Assim, passou-se pelos tópicos arrumando as palavras manualmente. Com as palavras normalizadas, submeteram-se, por meio de um *script*, os tópicos ao *Palmetto*⁸, coletando o resultado para cada métrica e armazenando em um arquivo *csv*.

5.2 Resultados

Nesta seção apresentam-se os resultados obtidos, dadas as extrações de tópicos e aplicação de métricas. A partir da execução do LDA, foram obtidos os tópicos com as palavras, e sua respectiva probabilidade de ocorrência no tópico, como pode ser visto na Tabela 5.3. Os rótulos foram atribuídos a partir de uma análise das *top-10* palavras do tópico, contando com a colaboração de outros usuários.

Tabela 5.3 – Top-5 tópicos de uma execução com $k = 50$.

"Maps"	"Spring Cloud"	"Android"	"Web"	"Git"
option*(0.178)	field*(0.192)	project*(0.114)	html*(0.165)	page*(0.213)
refer*(0.165)	configure*(0.111)	install*(0.087)	edit*(0.135)	window*(0.137)
custom*(0.148)	valid*(0.110)	build*(0.077)	javascript*(0.109)	import*(0.101)
active*(0.096)	filter*(0.087)	service*(0.068)	jquery*(0.064)	local*(0.055)
address*(0.079)	account*(0.063)	android*(0.052)	section*(0.061)	commit*(0.035)
angular*(0.069)	spring*(0.053)	package*(0.050)	page*(0.055)	branch*(0.033)
wonder*(0.037)	config*(0.043)	device*(0.033)	ajax*(0.049)	merge*(0.032)
year*(0.035)	cloud*(0.030)	step*(0.030)	integred*(0.039)	storage*(0.023)
adapt*(0.024)	schema*(0.028)	work*(0.022)	area*(0.032)	pull*(0.019)
camera*(0.021)	profile*(0.026)	studio*(0.020)	team*(0.023)	master*(0.019)

A Tabela 5.3 apresenta *top-5* tópicos de uma execução com $k=50$. Através das palavras

⁸ <http://palmetto.aksw.org/palmetto-webapp/>

de cada tópico, foi possível avaliar que o primeiro refere-se a algo sobre mapas, o segundo sobre a ferramenta *Spring cloud*, o terceiro e quarto sobre programação *Android* e *Web*, respectivamente, e o quinto trata de controle de versão.

Em uma análise geral sobre os tópicos obtidos, é possível inferir os assuntos mais recorrentes. Para a base de dados utilizada nesse trabalho, verifica-se, assim como nos tópicos acima, que predominam assuntos ligados à programação web e *mobile*, como *javascript*, *jquery*, *json*, *bootstrap*, *android*, além do controle de versão, programação orientada a objetos, servidores e banco de dados. Também verificam-se algumas ferramentas mais específicas, como o *Spring Cloud* e *Microsoft Azure*.

Para a avaliação do número de tópicos k , tendo em vista a variação das probabilidades dentro de um conjunto de tópicos, foram considerados vários cenários para avaliação dos tópicos com o uso das métricas. Para cada rodada de k , foram selecionados os *top-5*, *top-10*, *bottom-5* e *bottom-10* tópicos, e realizadas as médias para cada métrica, sempre com as *top-10* palavras de cada tópico.

Tabela 5.4 – Média *top-5* tópicos de cada k .

tópicos métrica	50	100	150	200
C_P	$0,0637 \pm 0,0507$	$-0,1036 \pm 0,0728$	$-0,0856 \pm 0,0798$	$-0,0787 \pm 0,1094$
C_V	$0,3711 \pm 0,0107$	$0,3858 \pm 0,0190$	$0,3812 \pm 0,0139$	$0,3844 \pm 0,0199$
C_{UCI}	$-0,8897 \pm 0,5023$	$-1,7497 \pm 0,8648$	$-1,6926 \pm 0,5220$	$-1,5257 \pm 0,5592$
C_{Umass}	$-3,2659 \pm 0,3837$	$-4,1342 \pm 0,8059$	$-3,5527 \pm 0,6668$	$-3,9029 \pm 1,0516$
C_{NPMI}	$-0,0178 \pm 0,0230$	$-0,0543 \pm 0,0326$	$-0,0528 \pm 0,0246$	$-0,0465 \pm 0,0224$
C_A	$0,1187 \pm 0,0040$	$0,1095 \pm 0,0103$	$0,1032 \pm 0,0088$	$0,1055 \pm 0,0119$

Na Tabela 5.4, para os *top-5*, $k = 50$ obteve melhores resultados em 5 métricas, somente $k = 100$ obteve melhor resultado na métrica C_V . Por outro lado, $k = 100$ obteve os resultados mais baixos em 4 métricas.

Tabela 5.5 – Média *top-10* tópicos de cada k .

tópicos métrica	50	100	150	200
C_P	$0,0497 \pm 0,0352$	$-0,0460 \pm 0,0754$	$-0,0971 \pm 0,0635$	$-0,0765 \pm 0,0555$
C_V	$0,3714 \pm 0,0039$	$0,3822 \pm 0,0153$	$0,3967 \pm 0,0232$	$0,3912 \pm 0,0200$
C_{UCI}	$-0,9023 \pm 0,3393$	$-1,5229 \pm 0,5657$	$-1,8050 \pm 0,5336$	$-1,5716 \pm 0,4373$
C_{Umass}	$-3,3976 \pm 0,4048$	$-4,0502 \pm 0,7220$	$-4,0608 \pm 0,9286$	$-4,0299 \pm 0,7523$
C_{NPMI}	$-0,0169 \pm 0,0147$	$-0,0435 \pm 0,0213$	$-0,0552 \pm 0,0207$	$-0,0475 \pm 0,0165$
C_A	$0,1208 \pm 0,0061$	$0,1135 \pm 0,0082$	$0,1046 \pm 0,0040$	$0,1055 \pm 0,0059$

Considerando os *top-10* tópicos, mais uma vez $k = 50$ obteve melhores resultados em

5 métricas, e $k = 150$ teve melhor resultado na métrica C_V , entretanto, ela também obteve os resultados mais baixos em outras 4 métricas, como pode ser visto na Tabela 5.5.

Tabela 5.6 – Média *bottom-5* tópicos de cada k .

tópicos métrica	50	100	150	200
C_P	$0,0741 \pm 0,0606$	$-0,0672 \pm 0,0929$	$-0,1398 \pm 0,1179$	$-0,1887 \pm 0,0396$
C_V	$0,3600 \pm 0,0036$	$0,3873 \pm 0,0103$	$0,3785 \pm 0,0279$	$0,4197 \pm 0,0188$
C_{UCI}	$-0,8318 \pm 0,5720$	$-1,6839 \pm 0,4421$	$-1,7090 \pm 0,5016$	$-2,1037 \pm 0,3356$
C_{Umass}	$-3,3056 \pm 0,2959$	$-3,6718 \pm 0,2346$	$-4,3252 \pm 0,6047$	$-4,7161 \pm 1,0846$
C_{NPMI}	$-0,0163 \pm 0,0256$	$-0,0532 \pm 0,0172$	$-0,0536 \pm 0,0200$	$-0,0661 \pm 0,0144$
C_A	$0,1177 \pm 0,0119$	$0,1053 \pm 0,0102$	$0,1000 \pm 0,0024$	$0,1035 \pm 0,0139$

De acordo com a Tabela 5.6, quando consideram-se os *bottom-5* (5 tópicos com menores valores em k), percebe-se o mesmo comportamento dos *top* tópicos, com $k = 50$ tendo números mais altos em 5 métricas, desta vez com $k = 200$ sendo melhor na métrica C_V . Por outro lado, $k = 200$ teve os menores resultados em 4 métricas.

Tabela 5.7 – Média *bottom-10* tópicos de cada k .

tópicos métrica	50	100	150	200
C_P	$0,0791 \pm 0,0310$	$-0,0830 \pm 0,0559$	$-0,1302 \pm 0,0686$	$-0,1616 \pm 0,0360$
C_V	$0,3624 \pm 0,0124$	$0,3993 \pm 0,0243$	$0,3822 \pm 0,0206$	$0,4142 \pm 0,0181$
C_{UCI}	$-0,7223 \pm 0,3689$	$-1,8525 \pm 0,5364$	$-1,6828 \pm 0,4890$	$-2,0122 \pm 0,3176$
C_{Umass}	$-3,2442 \pm 0,4796$	$-4,2511 \pm 0,7363$	$-4,2585 \pm 0,7395$	$-4,5108 \pm 0,6173$
C_{NPMI}	$-0,0098 \pm 0,0137$	$-0,0572 \pm 0,0197$	$-0,0527 \pm 0,0191$	$-0,0642 \pm 0,0119$
C_A	$0,1266 \pm 0,0093$	$0,1086 \pm 0,0097$	$0,1010 \pm 0,0028$	$0,0980 \pm 0,0037$

Conforme a Tabela 5.7, ao analisar os *bottom-10* tópicos, verificou-se mais uma vez que $k = 50$ foi melhor nas mesmas 5 métricas, com $k = 200$ tendo melhor resultado para C_V , entretanto, ele também apresentou resultados mais baixos em 4 métricas.

Tabela 5.8 – Média geral de cada k para todos os tópicos.

tópicos métrica	50	100	150	200
C_P	$0,0732 \pm 0,0192$	$-0,0562 \pm 0,0170$	$-0,0949 \pm 0,0173$	$-0,1210 \pm 0,0376$
C_V	$0,3685 \pm 0,0048$	$0,3850 \pm 0,0042$	$0,3913 \pm 0,0027$	$0,3978 \pm 0,0107$
C_{UCI}	$-0,7611 \pm 0,1341$	$-1,6295 \pm 0,1099$	$-1,7701 \pm 0,0620$	$-1,7125 \pm 0,2092$
C_{Umass}	$-3,3524 \pm 0,1603$	$-4,0884 \pm 0,1625$	$-4,0885 \pm 0,1146$	$-4,2988 \pm 0,4458$
C_{NPMI}	$-0,0099 \pm 0,0062$	$-0,0486 \pm 0,0042$	$-0,0548 \pm 0,0029$	$-0,0532 \pm 0,0083$
C_A	$0,1253 \pm 0,0033$	$0,1103 \pm 0,0030$	$0,1057 \pm 0,0029$	$0,1035 \pm 0,0040$

Para uma avaliação geral, foram observadas as médias para todos os tópicos de cada

valor de k , como pode ser visto na Tabela 5.8. Tal estudo seguiu na mesma linha das observações anteriores, com 5 métricas melhor avaliadas com $k = 50$, variando em C_V para $k = 200$, que também obteve os piores resultados para 3 métricas.

Tabela 5.9 – Aplicação do T-Test

T-Test	50 - 100	50 - 150	50 - 200	100 - 150	100 - 200	150 - 200
C_P	0,001061	0,000022	0,000704	0,049502	0,011509	0,318154
C_V	0,009099	0,000171	0,006700	0,066307	0,065934	0,205757
C_{UCI}	0,000874	0,000010	0,002340	0,109229	0,382064	0,612314
C_{Umass}	0,003592	0,000286	0,005443	0,999023	0,394317	0,377342
C_{NPMI}	0,000845	0,000009	0,001692	0,093030	0,243938	0,733183
C_A	0,005312	0,000306	0,000494	0,074479	0,051314	0,486185

Para uma comparação entre o número de tópicos, aplicou-se o *student T-Test*, gerando a Tabela 5.9. A partir do *T-Test*, com base na Tabela 5.8, verificou-se a superioridade de $k = 50$ em relação aos demais valores de k em 5 métricas, sendo inferior os demais somente na métrica C_V . Ao comparar os valores de k 100 com 150, não se pode afirmar algo para as métricas C_{UCI} e C_{Umass} , pois em ambas o valor de α é maior que 0.1. O mesmo vale para 100 com 200, incluindo a C_{NPMI} . Com relação ao 150 com 200, nenhuma hipótese pode ser afirmada.

5.3 Considerações finais

Por fim, baseado nos resultados apresentados, constata-se que, dentre os valores analisados para k , as execuções com 50 tópicos demonstraram melhores índices de coerência. Tal afirmação é baseada nas métricas aplicadas, apresentando os melhores números em 5 das 6 métricas utilizadas na elaboração do trabalho, e confirmadas com a aplicação do *Student T-Test*. Além do bom desempenho geral, com $k = 50$ obtiveram-se também os melhores resultados nas análises dos *top-n* e *bottom-n* tópicos, apresentando regularidade nos resultados.

Além de observar o número de tópicos e mensurar sua coerência, foi possível rotular tópicos com base nas *top* palavras por eles apresentadas. Tal como na Tabela 5.3, no qual pode-se associar as palavras, quando combinadas, a um determinado assunto. Através dessa análise sobre os tópicos em geral, observaram-se os assuntos mais recorrentes para o período verificado, compreendendo, principalmente, tópicos nas áreas de programação web e *mobile*, passando por programação orientada a objetos, bancos de dados, controle de versão, e também a algumas ferramentas em específico, como o *Microsoft Azure*.

6 CONCLUSÃO

Neste trabalho de conclusão de curso, foi realizada uma análise exploratória sobre a base de dados do *Stack Overflow*. Para a viabilizar a execução do trabalho, foram realizadas etapas de pré-processamento, com a limpeza dos dados, aplicação do LDA para extrair os tópicos, e pós-processamento, com a normalização das palavras obtidas nos tópicos.

Para avaliação dos tópicos obtidos, foram analisadas as principais palavras de cada tópico, reconhecendo relações entre elas a fim de rotular o conjunto, identificando o tópico e relacionando-o a um determinado assunto. Assim, foram observados os tópicos mais recorrentes no período especificado, ligados principalmente a programação web, em torno do *javascript*, *mobile*, controle de versão, etc.

Além da análise sobre as palavras, utilizaram-se métricas de coerência, submetendo as *top-10* palavras de cada tópico, buscando-se mensurar a relação entre elas. Com isso, foram feitas comparações entre os números de tópicos, denotados por k , buscando-se identificar o melhor valor para representar os dados analisados. Neste caso, os melhores resultados foram obtidos com $k = 50$, impondo-se diante dos demais valores, quando foram verificados todos os tópicos, e também nos *top-5*, *top-10*, *bottom-5* e *bottom-10* tópicos.

6.1 Trabalhos futuros

A fim de abranger a análise realizada neste trabalho, podem ser realizadas análises mais profundas na base de dados do *Stack Overflow*, compreendendo a evolução dos tópicos ao passar do tempo, comparando, por exemplo, a variação dos tópicos mais recorrentes em uma ordem cronológica.

Além disso, pode-se elaborar uma avaliação dos tópicos por métricas que usem a base de dados analisada, buscando-se mensurar a coerência a partir dos documentos que geraram os tópicos, com o intuito de investigar e comparar com outros meios de validação, como o *Wikipedia*.

REFERÊNCIAS

- BARUA, A.; THOMAS, S. W.; HASSAN, A. E. What are developers talking about? an analysis of topics and trends in stack overflow. **Empirical Software Engineering**, [S.l.], v.19, n.3, p.619–654, 2014.
- BLEI, D. M. Probabilistic topic models. **Communications of the ACM**, [S.l.], v.55, n.4, p.77–84, 2012.
- BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **Journal of machine Learning research**, [S.l.], v.3, n.Jan, p.993–1022, 2003.
- BOLELLI, L.; ERTEKIN, S.; GILES, C. L. Topic and Trend Detection in Text Collections Using Latent Dirichlet Allocation. In: ECIR. **Anais...** [S.l.: s.n.], 2009. p.776–780.
- FALEIROS, T. d. P. **Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais**. 2016. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.
- FITELSON, B. A probabilistic theory of coherence. **Analysis**, [S.l.], v.63, n.279, p.194–199, 2003.
- LINARES-VÁSQUEZ, M.; DIT, B.; POSHYVANYK, D. An exploratory analysis of mobile development issues using stack overflow. In: WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 10. **Proceedings...** [S.l.: s.n.], 2013. p.93–96.
- ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta. **Anais...** ELRA, 2010. p.45–50. <http://is.muni.cz/publication/884893/en>.
- ROCHA, A. M. et al. Documentação automatizada de APIs com tutoriais gerados a partir do Stack Overflow. , [S.l.], 2016.
- RÖDER, M.; BOTH, A.; HINNEBURG, A. Exploring the space of topic coherence measures. In: ACM INTERNATIONAL CONFERENCE ON WEB SEARCH AND DATA MINING. **Proceedings...** [S.l.: s.n.], 2015. p.399–408.

Stack Overflow. **Topic models evaluation in Gensim - Stack Overflow**. 2017.

STEYVERS, M.; GRIFFITHS, T. Probabilistic topic models. **Handbook of latent semantic analysis**, [S.l.], v.427, n.7, p.424–440, 2007.

TREUDE, C. et al. Programming in a socially networked world: the evolution of the social programmer. **The Future of Collaborative Software Development**, [S.l.], p.1–3, 2012.