Atividade 3: clusters de News Headlines

Jonlenes Castro*
Camila Moura†
Anderson Rocha‡

Resumo

O objetivo deste trabalho é encontrar quantidade mais adequada de clusters dado um conjunto de manchetes, sendo utilizado o algorítimo K-Means para clustering e PCA e t-SNE para visualização.

1. Introdução

Neste trabalho explorou-se o método de aprendizado de máquina não supervisionado para realizar o agrupamento do conjunto de manchetes.

Tendo em visa o tipo dos dados, é necessário realizar o pré processamento dos textos a fim de extrair os *features*, que serão futuramente usados no algorítimo de clusterização.

Nas seções seguintes serão apresentados o conjunto de dados que será utilizado nesta atividade e os procedimentos realizados durante o processo de agrupamento.

2. Atividades

Para identificar o numero mais adequado de *clusters*, foram realizadas e apresentadas as seguintes atividades:

- Pré processamento do *dataset* e extração de *Feature*;
- TF-IDF, K-mens e a escolha de K;
- Experimentos.

3. O *dataset*

A Million News Headlines é o *dataset* utilizado para este trabalho [1]. Este contém 1 milhão de dados de notícias publicadas durante um período de 15 anos pela fonte de notícias australiana ABC (Australian Broadcasting Corp).

*Instituto de Computação, Universidade de Campinas (Unicamp). **Contact**: jonlenes.castro@ic.unicamp.br
†Instituto de Computação, Universidade de Campinas (Unicamp). **Contact**: camila.moura@ic.unicamp.br
‡Instituto de Computação, Universidade de Campinas (Unicamp). **Contact**: anderson.rocha@ic.unicamp.br

Será utilizado inicialmente uma amostra de 10% deste *dataset* (100.000 exemplos), selecionados aleatoriamente e em seguida os dados serão divididos por ano (Conforme Tabela 1), sendo trabalhado cada ano separadamente.

Tabela 1. *Dataset* dividido por ano

Ano	Total de Exemplos	Ano	Total de Exemplos
2003	59343	2011	69919
2004	65975	2012	78547
2005	66320	2013	81016
2006	61568	2014	73361
2007	69431	2015	70004
2008	71591	2016	52162
2009	68867	2017	44182
2010	67715		

Esta amostra será utilizada no primeiro processamento que consiste na preparação dos dados para a extração de *features*, sendo discutido na seção a seguir.

4. Pré-processamento

Nesta etapa foi necessário realizar algumas alterações básicas do *dataset*, tais como:

- Conversão para *lowercase*: conversão de todos os caracteres do *dataset* para *lowercase*;
- Remoção de pontuação: todas as pontuações foram removidas utilizando a lista disponibilizada em [2];
- Remoção de palavras por tamanho: todas as palavras de tamanho menor ou igual a 2 foram removidas;
- *Stemming*: processo de reduzir as palavras flexionadas (ou, às vezes, derivadas) para a palavra raiz , base ou forma raiz - geralmente uma forma de palavra escrita. [3]. Este processo foi realizado em todo o *dataset* utilizando a NFTL [4]
- Remoção de números: todos os caracteres numéricos foram removidos.

5. *Feature Extration*

O próximo passo consiste no processo de extração de features dos arquivos de texto. Para isso foi utilizada a técnica de *Bag of Words*, onde é desconsiderando a gramática e a ordem das palavras, mas mantendo a multiplicidade [5]. Esta abordagem, olha-se o histograma das palavras dentro do texto, ou seja, considerando cada palavra contada como uma característica.

Essa foi a estrategia utilizada para a extração de features neste trabalho, utilizando a sklearn [6] para realizar essa contagem.

6. TF-IDF

Um problema com a utilização da frequência de palavras é que algumas palavras que aparecem com muita frequência começam a dominar no documento, mas podem não conter tanta "informação"em comparação com palavras menos frequentes [7].

Uma abordagem é redimensionar a frequência das palavras pela frequência com que aparecem em todos os documentos, de modo que as pontuações para palavras frequentes, que também são frequentes em todos os documentos, sejam penalizadas. Essa abordagem para pontuação é chamada de Frequência de Termo - Frequência de Documento Inversa (TF-IDF) [7].

Neste trabalho foi aplicada a IDF após toda extração de features utilizando a sklearn [6].

7. K-Means

K-Means clustering faz parte do grupo de método de aprendizado não supervisionado, que é utilizado quando é necessário analisar dados que não contém labels [8].

Este algoritmo encontra grupos nos dados, funcionando iterativamente atribuindo cada ponto de dados a um dos grupos K com base nas características fornecidas [8].

Para a sua execução é necessário informa-lo o valor de K desejado, pois o mesmo não o encontra sozinho. Nesse momento, também é necessário realizar a escolha de K, conforme apresentado na próxima seção.

8. Métodos para a escolha de K

Neste trabalho foram utilizados dois métodos para escolher o valor de K, sendo eles o método de Elbow (Seção 8.1) e a análise de Silhouette (Seção 8.2) .

8.1. O método de Elbow

A ideia do método Elbow é executar o agrupamento K-Means no conjunto de dados para um intervalo de valores de K e para cada um dos valores calcular a função de custo [9].

Próximo passo consiste em plotar um gráfico de *cost function* vs valor de K. Se o gráfico de linha tiver a forma de um braço, então o cotovelo (elbow) no braço é um possível valor para K [9].

O objetivo é obter um valor pequeno para a função de custo, mas o custo tende a diminuir para 0 à medida que aumenta o numero K. Então, deve se escolher um pequeno valor de K que ainda tenha um custo baixo, e o cotovelo geralmente representa onde se começa a ter valores decrescentes aumentando K [9].

8.2. Análise de Silhouette

A análise de *Silhouette* é uma maneira de medir a proximidade de cada ponto em um *cluster* com os pontos em seus *clusters* vizinhos [10].

Isso pode ser utilizado para descobrir o valor ideal para k durante o agrupamento *K-Means*. Os valores de *Silhouette* estão no intervalo de [-1, 1], onde um valor +1 indica que o exemplo está longe de seu *cluster* vizinho e muito próximo ao *cluster* que está designado. Da mesma forma, o valor -1 indica que o ponto está próximo ao *cluster* vizinho do que ao *cluster* que está designado, e um valor de 0 significa que está no limite da distância entre os dois *clusters*. Portanto, quanto maior o este valor, melhor é a configuração do *cluster* [10].

9. PCA e t-SNE

Uma etapa importante do processo de *clustering* é visualização dos resultados. Neste trabalho foi utilizada uma combinação de *Principal component analysis* (PCA) e *T-Distributed Stochastic Neighbouring Entities* (t-SNE) para isso.

O PCA é uma técnica para reduzir o número de dimensões em um conjunto de dados retendo a maioria das informações. Ele tenta fornecer um número mínimo de variáveis que mantém a quantidade máxima de variação ou informações sobre como os dados originais são distribuídos [11]. Neste caso, quando o numero de dimensões é reduzido para algo que possa ser plotado, pode ser utilizado para fazer a visualização dos resultados.

O t-SNE é outra técnica para redução de dimensionalidade e é particularmente adequada para a visualização de conjuntos de dados de alta dimensão, sendo que este utiliza uma técnica probabilística para fazer a redução [11].

No entanto, t-SNE é computacionalmente pesado, causando algumas limitações ao uso dessa técnica. Por exemplo, uma das recomendações (Conforme [11]) é que, no caso de dados dimensionais muito altos, talvez seja necessário aplicar outra técnica de redução de dimensionalidade antes de usar o t-SNE [11].

Então, nesse trabalho foi utilizado o PCA para encontrar as 20 melhores componentes, e estes foram utilizando pelo

t-SNE para obter o resultado final.

10. Experimentos com todo o dataset

Os experimentos realizados desconsideram algumas palavras do vocabulário no processo de extração de *features*, conforme lista abaixo:

- Desconsideração das *stopwords*;
- Desconsideração de palavras que aparece em mais 50% dos documentos;
- Desconsideração de palavras que não ocorrem em pelo ao menos 5 documentos, para 1-gram, indo até 2 documentos para 4-gram (com o aumento do gram, a quantidade de repetição nos documentos diminui);

10.1. Experimento 1

A quantidade de *features* utilizada inicialmente foi 500 para a execução do primeiro teste, com 10% do *dataset*. Esse experimento foi realizado utilizando 1-gram, 2-gram, 3-gram e 4-gram para a extração de *features*. Após a extração foi realizado o agrupamento com *K-means* para valores de K variando de 2 a 250, conforme pode ser analisada nas Figuras 1 e 2.

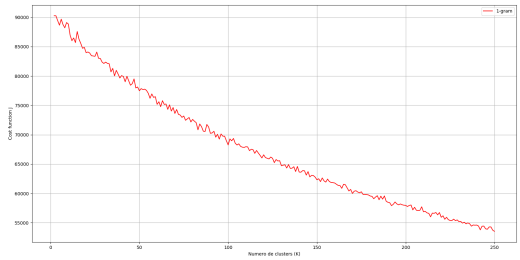

Figura 1. Cost function vs. K (1-gram)

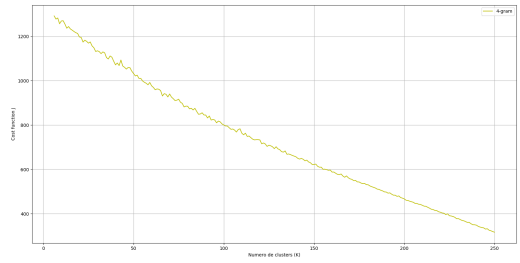

Figura 2. Cost function vs. K (4-gram)

Os gráficos para 1-gram até 4-gram mantiveram uma curvatura similar, sendo a principal diferença o valor da função de custo, que para o 1-gram está na casa de 10^5 e no 4-gram em 10^3

Para analisar melhor eles valores, foi realizada a plotagem desses valores com K variando de 10 em 10 para encontrar esses pontos, utilizando o método de *Elbow* pode se encontrar diversos valores para K, como será analisado ainda nesta seção.

No mesmo teste também foi realizado a analise de *silhouette*, os gráficos para o 1-gram e 4-gram podem ser visto nas Figuras 3 e 4, respectivamente.

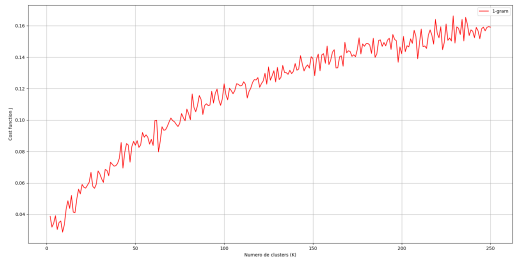

Figura 3. Silhouette vs. K (1-gram)

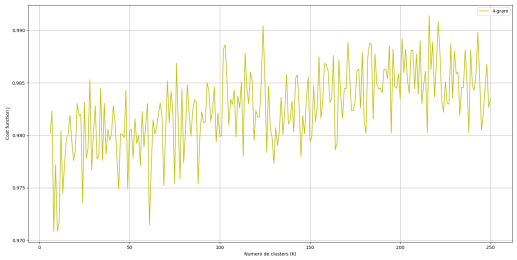

Figura 4. Silhouette vs. K (4-gram)

Como pode ser analisado no gráfico, os valores *silhouette* aumentaram com o aumento da quantidade de termos (gram), onde tem-se valor máximo de 0.16 para o 1-gram e de 0.99 para o 4-gram.

Um ponto importante já pode ser analisada aqui, com 4-gram tem-se os menores valores para a função de custo e os maiores valores para *silhouette*, indicando ser um bom candidato para uma boa extração de *features*.

10.2. Experimento 2

Com o intuindo de melhor o agrupamento, foi aumentada a quantidade de *features* para 1000 e diminuída a quantidade de K testados $(2 <= K <= 100)$ para melhor visualização do gráfico e diminuição do tempo de execução. Os resultados para 3-gram podem ser visto na Figura 5.

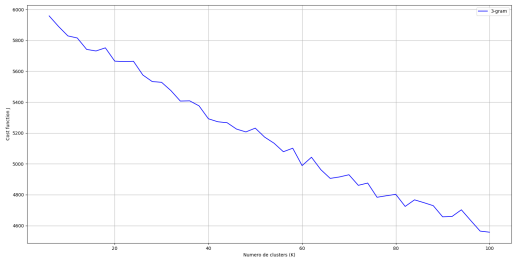

Figura 5. Cost function vs. K (3-gram)

Novamente obteve-se diversos pontos possíveis para K, sendo alguns deles para k menor do 20. Para a finalização desta parte será realizado um ultimo experimento, como se segue.

10.3. Experimento 3

Neste experimento foram realizadas mais algumas alterações no *dataset*, como se segue:

- Remoção de *headlines* duplicadas: ao analisar o *dataset* é possível perceber que se tem muitos exemplos duplicas, removendo-os, o tamanho do *dataset* é reduzido para 643715 exemplos;
- Todo o *dataset* foi utilizado (sem os valores duplicados);
- Foram considerados 10.000 *features*;
- *K-Means* foi executado somente para $2 <= K <= 20$.

Os resultados obtidos para 4-gram podem ser visto nas Figuras 6 e 7.

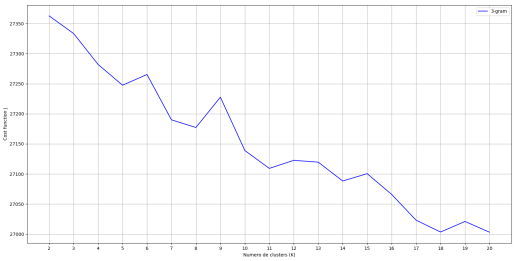

Figura 6. Cost function vs. K (4-gram) - 1000 features

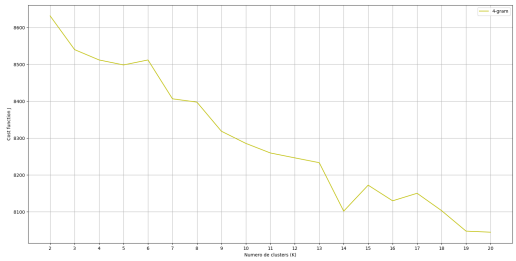

Figura 7. Cost function vs. K (4-gram) - 10000 features

Nessa primeira analise, considerando a função de custo e análise *silhouette* obteve se os melhores resultados com utilizando todos os exemplos não duplicados no *dataset*, 1000 *features* e 4-gram. No entanto, apareceu cotovelo em quase todos os pontos, quando se modificava o modelo treinado.

É percebível, que o numero de *clusters* exato dependo do objetivo e do que se deseja extrair do dataset. Neste caso, foi utilizado dois critérios para selecionar um possível numero de *clusters*: considerando todos os testes, os cotovelos mais apareceram e os que tiveram o melhor valor de *silhouette*.

Na Tabela 2 abaixo são apresentados os valores que mais apareceram em cada um dos casos (considerando valores de K menores do que ou igual a 20):

Tabela 2. Valores para K

Grams	Valores de K
1-gram	4, 6, 9, 10, 12, 13, 15, 16, 18, 19
2-gram	2, 3, 4, 5, 7, 8, 10, 11, 12, 15, 17, 19
3-gram	3, 5, 8, 9, 12, 14, 16, 17
4-gram	5, 7, 8, 9, 11, 14, 16, 18, 19

Alguns dos valores que mais apareceram foi: 5, 12. Para esses valores de K, foi executada novamente o treinamento e obtido os valores de *silhouette* utilizando 1-gram e 4-gram, conforme Tabela 3.

Tabela 3. Silhouette

Gram	K	Silhouette
1-gram	5	0.0154
1-gram	12	0.0233
4-gram	5	0.9719
4-gram	12	0.9761

Como pode ser observado, os valores de *Silhouette* foram bem melhores para 4-gram, sendo praticamente equivalente para K igual a 5 e K igual a 12. Nesse caso foi utilizando $K = 5$ para apresentar as palavras comuns em cada *clusters*, conforme abaixo (serão mostradas apenas 10 palavras como exemplo para cada *cluster*).

- Cluster 1: vic countri hour podcast, rural qld rural report, rural nsw rural report, sa countri hour podcast,

nation rural news wednesday, man face court accus, aung san suu kyi, test day live blog, nation rural news tuesday, speak abc news breakfast;

- Cluster 2: nation rural news monday, youth mental health servic, form guid men m, gold coast man face, gold coast light rail, gold coast hit run, gold coast high rise, gold coast commonwealth game, given suspend jail term, given good behaviour bond;
- Cluster 3: nsw countri hour wednesday, countri hour wednesday decemb, countri hour wednesday septemb, countri hour wednesday novemb, countri hour wednesday april, countri hour wednesday june, countri hour wednesday octob, countri hour wednesday januari, countri hour wednesday august, countri hour wednesday februari;
- Cluster 4: tas countri hour friday, countri hour friday octob, countri hour friday april, countri hour friday june, countri hour friday august, countri hour friday juli, countri hour friday march, countri hour friday novemb, heavi rain caus flood, heavi rain caus flash;
- Cluster 5: wa countri hour podcast, countri hour podcast octob, countri hour podcast th, countri hour podcast decemb, countri hour podcast septemb, countri hour podcast juli, countri hour podcast august, countri hour podcast februari, countri hour podcast januari, countri hour podcast march.

11. Experimentos por ano

Para os experimentos por ano foi considerado todo o dataset (com 1M de exemplos), dividido por ano, conforme Seção 3, e posteriormente, para cada ano foi removidas headlines duplicadas. Após a remoção das headlines duplicadas, para cada ano, os exemplos diminuíram de 34% a 37%.

Após esse processo, os mesmo experimentos da seção anterior foi realizado para cada ano, conforme resumo dos experimentos apresentados na Tabela 4:

Tabela 4. Experimentos por ano

Ano	K	Ano	K
2003	3, 5, 8, 11, 16	2011	3, 5, 11
2004	3, 6, 9, 11, 15	2012	3, 5, 8, 13, 16
2005	4, 6, 9, 11, 13, 18	2013	4, 7, 10, 13, 15
2006	3, 8, 11, 13, 15	2014	3, 7, 10, 14
2007	3, 5, 6, 10, 18	2015	5, 8, 11, 14, 16
2008	4, 8, 13, 16, 19	2016	4, 6, 10, 13, 17
2009	4, 9, 13, 15, 18	2017	4, 8, 10, 12, 15, 17

Com a divisão do *dataset* por ano, facilitou a utilização do algoritmo de agrupamento, pois a quantidade de dados é bem menor com essa divisão. No entanto, os resultados com o *dataset* divididos por ano não foi tão bons quanto o do *dataset* inteiro, considerando o valor de *silhouette*, e principalmente na visualização dos resultados.

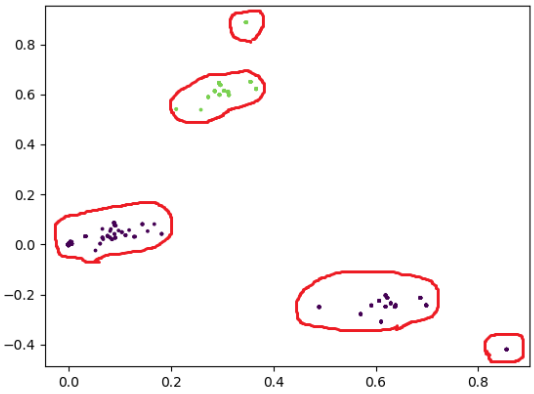

Figura 9. Visualização de 5 clusters

Considerando a Tabela 4 é possível perceber que o valor de $K = 3$ aparecer em quase todos os anos, sendo um possível escolha para valor de K.

12. Visualização dos clusters

Segue alguns dos gráficos montados no decorrer dos experimentos desse trabalho, demostrando como os estão agrupados visualmente, conforme Figura 8 que mostra o valor $K = 3$, conforme mencionado na Seção 11 e na Figura 9 apresenta os 5 *clusters*, conforme Seção 10. No geral pode ser visualizados 3 grandes grupos e mais alguns pequenos grupos,

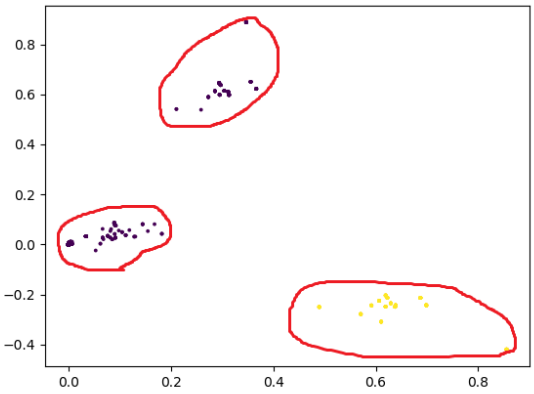

Figura 8. Visualização de 3 clusters

13. Conclusão e Trabalhos Futuros

O número de *clusters* encontrados não é necessariamente o melhor, nem o absoluto, pois o numero de *clusters* "adequado" pode depender do objetivo do agrupado.

Neste trabalho foi escolhido alguns valores para K baseado na função de custo e analise de *silhouette*. Existe outras analises e diversos experimentos que ainda podem ser feito com este *dataset* para aprimorar ainda mais o valor de K.

Referências

[1] Rohit Kulkarni. A million news headlines - abc australia 2003-2017, 2018. 1

[2] R Python Software Foundation. Common string operations, 2012. Disponível em: https://docs.python.org/3.1/library/string.html. 1

[3] Wikipédia. Stemming, 2018. Disponível em: https://en.wikipedia.org/wiki/Stemming. 1

[4] NLTK Project. Natural language toolkit, 2018. Disponível em: https://www.nltk.org/#natural-language-toolkit. 1

[5] Jason Brownlee. A gentle introduction to the bag-of-words model, 2017. Disponível em: https://machinelearningmastery.com/gentle-introduction-bag-words-model/. 2

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 2

[7] Julia Silge and David Robinson. Term frequency and inverse document frequency (tf-idf), 2018. Disponível em: https://cran.rproject.org/web/packages/tidytext/vignettes/tf_idf.html. 2

[8] ANDREA TREVINO. Introduction to k-means clustering, 2016. Disponível em: https://www.datascience.com/blog/k-means-clustering. 2

[9] Robert Gove. Using the elbow method to determine the optimal number of clusters for k-means clustering, 2017. Disponível em: https://bl.ocks.org/rpgove/0060ff3b656618e9136b. 2

[10] KAPILDALWANI. Using silhouette analysis for selecting the number of cluster for k-means clustering, 2015. Disponível em: https://kapilddatascience.wordpress.com/2015/11/10/using-silhouetteanalysisforselectingthenumberofclusterforkmeansclustering/. 2

[11] Luuk Derksen. Visualising high-dimensional datasets using pca and t-sne in python, 2016. Disponível em: https://medium.com/@luckylwk/visualisinghighdimensionaldatasetsusingpcaandtsneinpython8ef87e7915b. 2