

# Relatório do Trabalho de Inteligência Artificial

Arthur Um Augusto Calado Bueno Fernando Chiu Hsieh João Trevisan Martins Lucas Shie Lai  
No USP: 9778898 No USP: 9779134 No USP: 9436743 No USP: 9778599 No USP: 9778710  
arthur.um@usp.br augusto.bueno@usp.br fernando.hsieh@usp.br joao.trevisan.martins@usp.br lucas.lai@usp.br

## I. PRELIMINARES

O presente relatório descreve o trabalho semestral de Inteligência Artificial realizado pelo nosso grupo. O objetivo do trabalho foi o de realizar uma experimentação com algoritmos de agrupamento e, para tal, utilizamos os algoritmos K-means, K-means++, X-means e *Self Organizing Maps* (SOM).

A implementação desses algoritmos foi feita utilizando a linguagem de programação *Python*, e a amostragem dos dados utilizando *Java*. A nossa decisão de implementar os códigos em *Python* se baseou na ampla variedade de bibliotecas matemáticas e de aprendizado de máquina disponíveis livremente e na facilidade de implementação que elas nos proporcionam. As bibliotecas que utilizamos foram as seguintes:

- 1) nltk [1] para a seleção de stopwords;
- 2) matplotlib [2] para a visualização de dados;
- 3) numpy [3] para operações matemáticas;
- 4) scikit [4] para operações matemáticas e pré processamento;
- 5) joblib [5] para implementação de paralelismo;
- 6) somoclu [6] para a implementação do SOM;
- 7) pandas [7] para a manipulação dos conjuntos de dados;
- 8) wordcloud [8] para a geração das nuvens de palavras.

A nossa implementação dos algoritmos pode ser encontrada nas pastas anexadas ou no nosso repositório online<sup>1</sup>. Os experimentos realizados utilizaram os conjuntos de dados BBC [9] e Reuters [10], além da biblioteca *scikit* para pré-processá-los.

O nosso grupo se deparou, principalmente, com uma questão de tempo de processamento e de *outliers*, os quais são explicados mais detalhadamente nas seções III e V respectivamente.

## II. CONJUNTO DE DADOS

Os conjuntos de dados escolhidos por nosso grupo foram o BBC [9] e o Reuters [10], ambos referentes a textos de notícias em inglês. O primeiro córpus é composto por dois mil duzentos e vinte e cinco documentos rotulados de acordo com 5 categorias de notícias - *business*, *entertainment*, *politics*, *sport*, *tech*, como explicitado na Tabela 1.

O segundo córpus possui seis mil documentos também rotulados de acordo com cinco categorias - *Exchanges*, *Orgs*, *People*, *Places*, *Topics*-, contudo, devido a apresentação do dataset, o grupo não conseguiu acesso às categorias desse córpus, somente aos textos.

Tabela I  
QUANTDADE DE NOTÍCIAS DE CADA CLASSE DO CÓRPUS BBC

Classe	Quantidade de Documentos
<i>Business</i>	510
<i>Entertainment</i>	386
<i>Politics</i>	417
<i>Sport</i>	511
<i>Tech</i>	401

### A. Pré-Processamento

O grupo optou por realizar o pré-processamento do conjunto de dados para as representações Binário, *Term Frequency* (TF) e *Term Frequency Inverse Document Frequency* (TFIDF) e para uma representação distribuída de palavras (Word2vec) [11]. Além disso, nós decidimos também experimentar as técnicas *Latent Semantic Analysis* (LSA)<sup>2</sup> e Lematização<sup>3</sup>

O pré-processamento realizado deixou todas as palavras dos córpuses em caixa baixa e removeu as *stopwords* em inglês de acordo com a lista de *stopwords* da biblioteca NLTK para o inglês [1]. O procedimento de tokenização realizado separou as palavras de acordo com os espaços em branco.

Tendo em vista a quantidade de palavras distintas (atributos) dos córpuses ser alta -cerca de trinta mil palavras palavras - o grupo optou por considerar somente as três mil palavras mais frequentes a fim de otimizar o processamento e eliminar ruídos.

Em relação à representação Word2Vec, os *embeddings* treinados do Google News [14] foram utilizados para computar a representação dos documentos textuais, calculando a média dos vetores de cada texto. Assim, essa representação apresenta 300 atributos, que se referem à dimensionalidade dos *embeddings*. A utilização do LSA segue a parametrização sugerida pela documentação da biblioteca *scikit*, reduzindo a dimensionalidade das representações para cem componentes.

### B. Vizualização das Representações

A fim de ter uma noção preliminar sobre a distribuição dos dados em cada representação, utilizamos a técnica de redução

<sup>2</sup>O tratamento LSA [12] baseia-se em uma técnica de diminuição de dimensionalidade, *Singular Value Decomposition* (SVD), aplicada na matriz das representações consideradas. Essa diminuição visa na suavização dos ruídos além de uma acentuação de padrões presentes nos textos

<sup>3</sup>A técnica de lematização [13] trata palavras flexionadas de diferentes formas porém com a mesma raiz de maneira igual, essa técnica é relativamente custosa pelo fato de também levar em conta o contexto em que a palavra está incluída.

<sup>1</sup><https://github.com/Ferch42/Trabalho-de-IA>

de dimensionalidade TSNE [15] para poder visualizar os dados em três dimensões.

A primeira representação, ilustrada nas imagens 1 e 2, é a binária. Nessa representação, cada texto é tratado como um vetor, e cada posição desse vetor representa uma palavra que está presente no córpus inteiro. Caso um texto possua em seu conteúdo uma determinada palavra, seu vetor recebe o valor 1 na posição referente à palavra em questão. Caso contrário, o valor nessa posição continua como 0.

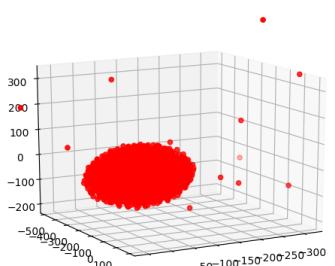


Figura 1. Esparsidade do córpus Reuters com a representação binária.

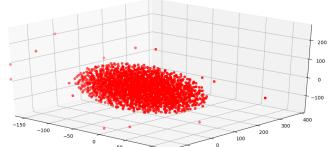


Figura 2. Esparsidade do córpus BBC com a representação binária.

Diferentemente da representação anterior, os vetores gerados pela representação TF, figuras 3 e 4, em vez de apenas assumirem valores 0 e 1 em suas posições baseado na ocorrência ou não de determinadas palavras, dessa vez são mantidos as frequências das palavras nas posições do vetor.

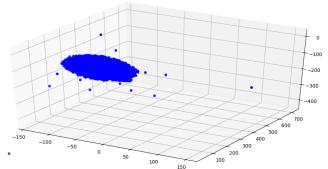


Figura 3. Esparsidade do córpus Reuters com a representação TF.

Já na representação TFIDF, figuras 5 e 6, não é simplesmente contada a quantidade de vezes que a palavra aparece em um determinado texto. Em vez disso, cada posição dos vetores é composta pela frequência de uma determinada palavra multiplicada pelo inverso da frequência desse termo no córpus. Dessa maneira, os atributos são ponderados por um valor que

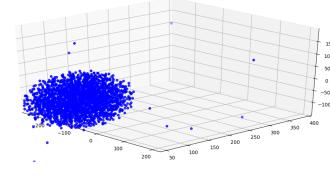


Figura 4. Esparsidade do córpus BBC com a representação TF.

diz respeito à relevância do atributo para discriminar um dado documento.

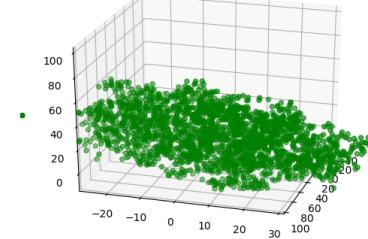


Figura 5. Esparsidade do córpus Reuters com a representação TFIDF.

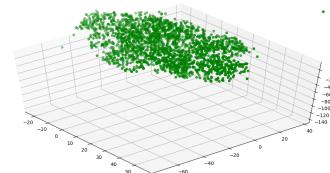


Figura 6. Esparsidade do córpus BBC com a representação TFIDF.

A técnica de representação Word2Vec, figuras 7 e 8, possui uma estratégia bem diferente das demais. Basicamente, um espaço multi-dimensional é criado e cada uma das palavras do córpus é representada por um vetor nesse espaço. Por fim, cada documento de texto assume a representação de um vetor com o valor da média dos vetores de todas as palavras que o compõem. Palavras que compartilham os mesmos contextos são posicionadas de maneira a ficar próximas nesse espaço.

Após a análise visual dos gráficos anteriores, nós notamos que as representações Binário e TF apresentam uma densidade de dados grande em relação à origem e que os dados não se distanciam muito um dos outros. Em contrapartida, as representações TFIDF e Word2Vec apresentam um espalhamento dos dados maior. Assim, conseguimos supor que a tarefa de clusterização obterá resultados melhores para as representações TFIDF e Word2Vec, mesmo levando em consideração as distorções que o algoritmo de redução de dimensionalidade traz.

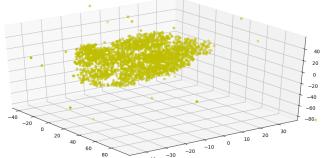


Figura 7. Esparsidade do córpus Reuters com a representação word2vec.

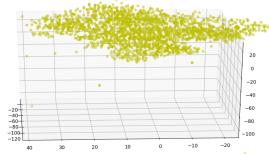


Figura 8. Esparsidade do córpus BBC com a representação word2vec.

### III. AMOSTRAGEM

Para que pudéssemos definir quais seriam as melhores configurações, tanto para o uso das diferentes técnicas de clusterização quanto para o uso da rede neural SOM, optamos por fazer um treino exploratório a fim de determinar os melhores parâmetros para os nossos algoritmos, ranqueando e ordenando os experimentos pelo resultado do índice *Silhouette*, escolhendo somente os melhores para o pós-processamento.

Devido a testes realizados, percebemos que cada iteração do algoritmo *K-means* levaria aproximadamente cinco minutos e cada iteração do algoritmo SOM tomaria cerca de meia hora. Com isso, concluímos que caso testássemos todos os parâmetros que desejávamos, o tempo que isso levaria seria exorbitante.

Para contornar essa situação, tomamos a decisão de amostrar 40% de ambos os córpus antes de dar início a fase de testes. Dessa maneira, não estaríamos tão sujeitos a perda de informação, uma vez que esta é uma quantidade significativa de textos e, ao mesmo tempo, teríamos a opção de conseguir testar mais diferentes configurações de nossos algoritmos, já que todos os testes demorariam muito menos para terminar.

No caso do Reuters, que possui seis mil arquivos de texto, utilizamos um critério de seleção aleatória de dois mil e quatrocentos arquivos. Tendo em vista que o córpus BBC encontrava-se rotulado, realizamos processo de amostragem levando em conta a proporção das cinco classes que o compunham.

### IV. PROCESSAMENTO

Baseado no teorema do limite central, que afirma que ao utilizarmos um número de experimentos igual a trinta, a distribuição dos resultados se aproxima da distribuição normal, decidimos realizar os experimentos para cada configuração do *K-means*, *K-means++* e *X-means* trinta vezes. Dessa maneira,

a aleatoriedade dos algoritmos anteriores não seriam prejudiciais para as análises posteriores dos resultados. Para o SOM realizamos cinco experimentos para cada configuração, por conta da demasiada demora na execução do algoritmo.

#### A. K-means e Variações

Os requisitos para as métricas de distância no *K-means* são distância euclidiana e similaridade cosseno. O grupo acrescentou a distância *manhattan* com o intuito de descobrir diferentes resultados que possam enriquecer e diversificar as análises dos agrupamentos. Dentre os parâmetros considerados para o *K-means* e suas variações, temos:

- ‘LSA’ : se os dados foram pré-processados com a técnica LSA ou não;
- ‘córpus’: ‘bbc’ ou ‘reuters’;
- ‘distância’: algoritmo utilizado para calcular as distâncias;
- ‘inicialização’: algoritmo de inicialização dos centróides de cada cluster;
- ‘ncluster’: número de clusters (variando de 2 a 7);
- ‘processamento’: se os dados foram lematizados ou não;
- ‘representação’: tipo de representação dos dados.

*1) Normal:* O *K-means* implementado na sua forma padrão inicializa os dados de forma aleatória, sem nenhuma consideração sobre as suas coordenadas individuais no espaço vetorial dos dados. São escolhidos pontos aleatórios do conjunto de dados para serem os centróides.

*a) Resultados K-means padrão e melhores parâmetros para córpus BBC:* Os melhores resultados de acordo com o índice silhueta para os testes com *K-means* inicialização padrão para o córpus BBC são os seguintes:

Tabela II  
MELHORES PARAMETRIZAÇÕES K-MEANS PARA O CÓRPUS BBC

Sil	LSA	dist	ncluster	proc	representação
0.165	False	Manhattan	2	Lemma	TF
0.163	True	Euclidiana	2	Lemma	TF
0.161	True	Manhattan	2	Lemma	TF

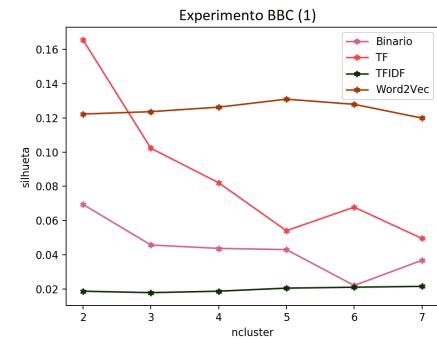


Figura 9.

b) Análise dos Parâmetros para Córpus BBC: Os parâmetros utilizados no experimento apresentado a seguir foram distância manhattan, sem tratamento LSA e processamento Lemma. Tal configuração de parâmetros gerou o resultado apresentado na figura 9.

A representação TF foi a responsável por atingir o maior valor de índice *Silhouette*, com número de clusters igual a 2. Notamos que os valores de silhueta para as representações Binário e TF tendem a diminuir com o aumento no número de clusters e que o desempenho do algoritmo para as representações Word2Vec e TFIDF manteve-se constante.

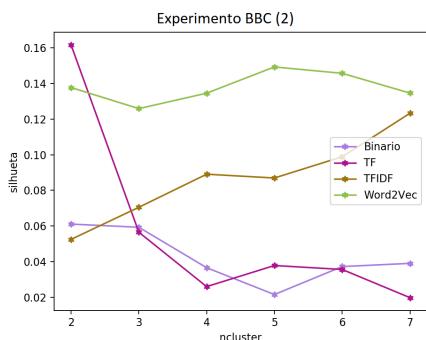


Figura 10.

Os parâmetros para o experimento 2 foram os mesmos do experimento anterior, exceto pelo fato de utilizarmos o tratamento LSA. Os resultados desse experimento mostram curvas semelhantes ao experimento 1, contudo com melhorias nas representações Word2Vec e TFIDF.

c) Resultados K-means padrão e melhores parâmetros para córpus Reuters: Os melhores resultados de acordo com o índice silhueta para os testes com *K-means* inicialização padrão para o córpus Reuters são os seguintes:

Tabela III

MELHORES PARAMETRIZAÇÕES K-MEANS PARA O CÓRPUIS REUTERS

Sil	LSA	dist	ncluster	proc	representação
0.327	True	Cosseno	2	Lemma	Word2Vec
0.325	True	Cosseno	2	Normal	Word2Vec
0.324	True	Euclidiana	2	Lemma	Word2Vec

d) Análise dos Parâmetros para Córpus Reuters: Os parâmetros utilizados no experimento apresentado a seguir foram distância cosseno, tratamento LSA e processamento Lemma. Tais condições de parâmetros geraram o resultado apresentado na figura 11.

Para o número de cluster igual a dois, obtém-se o melhor valor *Silhouette* utilizando a representação Word2Vec. Para certos números de clusters nota-se que as representações TFIDF e Word2Vec assumem os mesmos valores de *Silhouette*. Nota-se também que os valores *Silhouette* para o córpus Reuters tendem a ser maiores do que o do córpus BBC, o que indica uma melhor separação dos clusters.

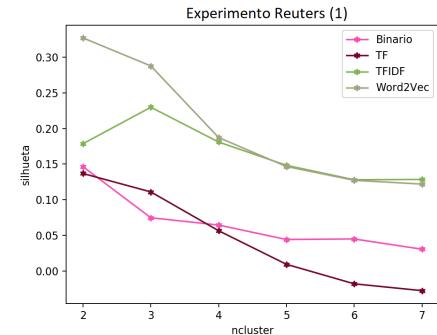


Figura 11.

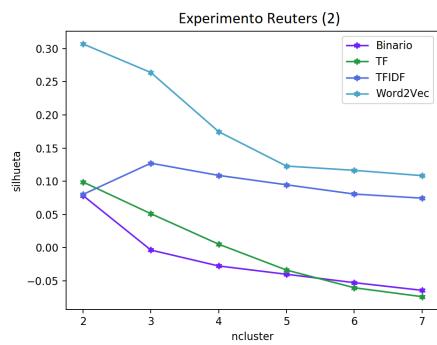


Figura 12.

Sobre os parâmetros do experimento 2 apresentado na figura 12, o que o difere do experimento um, é a não utilização do tratamento LSA.

Apontamos que o resultado do TFIDF para o experimento sem LSA foi pior do que no outro experimento, e que de maneira semelhante ao córpus BBC, as representações Word2Vec e TFIDF obtiveram bons resultados em comparação com as outras duas.

2) *K-means++*: A construção do *K-means++* seguiu os conceitos presentes nos *slides* disponibilizados em aula pela Profa. Dra. Sarajane M. Peres. A principal ideia por de trás do algoritmo construído pelo grupo, foi, em um primeiro momento, selecionar aleatoriamente os centróides via inicialização padrão. Após isso, para cada centróide gerado realizamos os seguintes cálculos:

- 1) calcula-se a distância de todos os dados para cada centróide;
- 2) para cada centróide computamos a soma dos quadrados das distâncias obtidas ( $\sigma_{total}$ );
- 3) para cada dado, determinamos a probabilidade de ele ser escolhido como novo centroid pela fórmula:  $(distancia_{dado\ centroid})^2 / (\sigma_{total})$ .

Tal sequência de passos proporciona um melhor espalhamento dos centróides pelos dados.

a) Resultados *K-means++* e melhores parâmetros: Os melhores resultados de acordo com o índice silhueta para os testes com *K-means* inicialização ++ são os seguintes:

Tabela IV  
MELHORES PARAMETRIZAÇÕES K-MEANS++ PARA O CÓRPUS REUTERS

Sil	LSA	dist	ncluster	proc	representação
0.329	True	Cosseno	2	Lemma	Word2Vec
0.328	True	Cosseno	2	Normal	Word2Vec
0.327	True	Euclidiana	2	Lemma	Word2Vec

Tabela V  
MELHORES PARAMETRIZAÇÕES K-MEANS++ PARA O CÓRPUS BBC

Sil	LSA	dist	ncluster	proc	representação
0.196	True	Euclidiana	7	Normal	TFIDF
0.179	True	Euclidiana	2	Normal	TF
0.175	True	Euclidiana	7	Lemma	TFIDF

b) Análise dos Parâmetros para Corpus BBC: O gráfico da imagem 13 trata do experimento que gerou o maior valor de índice *Silhouette* para o córpus BBC no algoritmo *K-Means++*. Nele os parâmetros foram distância euclidiana, processamento normal e tratamento LSA.

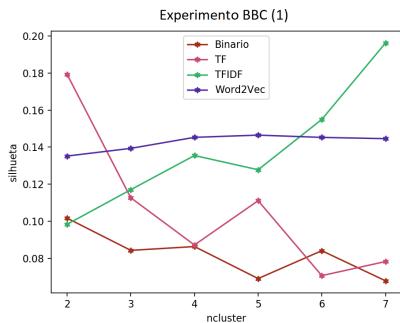


Figura 13.

Em uma análise inicial notamos que a representação TFIDF foi a responsável por atingir, para o algoritmo *K-Means++*, o maior valor de índice *Silhouette* (aproximadamente 0.196), para o número de clusters igual a 7. Outro destaque interessante é o fato da representação binária apresentar, para o número de cluster igual a dois, o segundo maior índice *Silhouette* dada a presente configuração de parâmetros.

Os parâmetros do experimento 2 foram os mesmos do experimento anterior, exceto pelo fato de não utilizarmos o tratamento LSA. O que nota-se de imediato é a diminuição evidente nos valores dos índices *Silhouettes* para todas as representações.

Percebe-se que para este experimento a representação Word2Vec se saiu melhor que as demais representações para todos os números de cluster. Em contrapartida a representação TFIDF que outrora saiu-se bem, no experimento atual possui os piores valores de *Silhouette*. Já as demais representações,

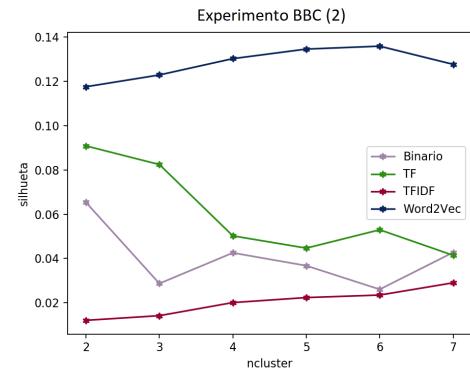


Figura 14.

aparentemente, seguiram um padrão semelhante ao do experimento anterior, ou seja, vão decaíndo conforme aumenta-se o número de cluster.

Notamos também que os resultados de *Silhouette* para a inicialização ++ tendem a ser superiores do que o da inicialização padrão para o BBC.

c) Análise dos Parâmetros para Córpus Reuters: O gráfico da figura 15 pertence ao experimento que apresentou o melhor valor de índice *Silhouette* para o córpus Reuters no algoritmo *K-Means++*, nele os parâmetros foram distância cosseno, processamento Lemma e tratamento LSA.

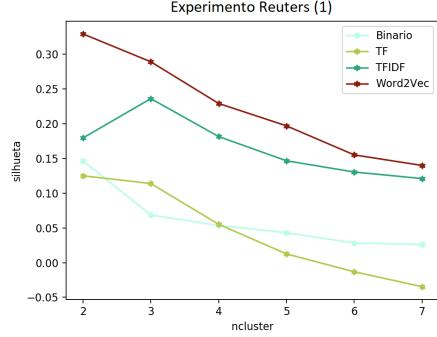


Figura 15.

A curva da representação Word2Vec durante todo o experimento obteve os melhores valores *Silhouette*. Logo em seguida a curva da representação TFIDF atingiu os segundos melhores resultados.

O parâmetro alterado no experimento mostrado na figura 16 foi apenas o LSA. No caso, o realizamos sem esse tratamento. Similar ao experimento anterior, as representações Word2Vec e TFIDF possuem, respectivamente, os melhores valores *Silhouette* para o presente experimento. Outra vez, o melhor *Silhouette* é o da representação Word2Vec para o número de cluster igual a dois.

3) *X-means*: A implementação do *X-means* foi baseada no artigo *X-means: Extending K-Means with Efficient Estimation*

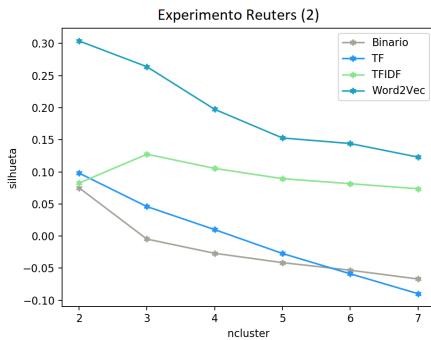


Figura 16.

of the Number of Clusters [16]. A lógica do algoritmo consiste em, basicamente, duas etapas. A primeira etapa executa um *K-means* sobre o conjunto de dados para que se gerem agrupamentos preliminares. A segunda etapa baseia-se, essencialmente, em realizar divisões nos clusters gerados na primeira etapa a fim de melhorar a qualidade do agrupamento. Como medida de avaliação para os agrupamentos gerados, utilizamos o índice *Silhouette*, também implementado pelo grupo na linguagem Python.

*a) Informações pertinentes do algoritmo X-means:*

A implementação do algoritmo *X-means* do grupo faz uso da nossa implementação de *K-means++* para realizar os agrupamentos a fim de alcançar melhores resultados. A quantidade de clusters inicial para os nossos experimentos foi 2 e a quantidade máxima de clusters foi 7.

*b) Resultados X-Means:* Um parâmetro que é configurado e alterado automaticamente no algoritmo *X-means* é o número de cluster. Por esse motivo, não apresentamos gráficos sobre o índice *Silhouette* com o aumento do número de cluster.

Tabela VI

MELHORES PARAMETRIZAÇÕES X-MEANS PARA O CÓRPUS REUTERS

Sil	LSA	dist	proc	representação
0.314	True	Cosseno	Normal	TFIDF

Tabela VII

MELHORES PARAMETRIZAÇÕES X-MEANS PARA O CÓRPUS BBC

Sil	LSA	dist	proc	representação
0.268	True	Euclidiana	Normal	Binário

Foi possível notar que o melhor resultado do *X-means* para o Reuters foi inferior aos dos melhores resultados do algoritmo *k-means* para ambas as outras inicializações. Já no caso do BBC, o *X-means* alcançou melhores resultados dos experimentos. Isso pode dever-se ao fato de o *X-means* sempre busca melhorar o valor silhouette cada vez que ele realiza a divisão de um cluster.

**4) Conclusão sobre os Means:** Levando em conta os gráficos e as análises dessa seção, além de outros estudos realizados pelo grupo que não estão aqui representados, realizamos uma série de considerações a respeito dos resultados, algoritmos e parâmetros da etapa de processamento.

- Durante as análises dos resultados, notamos que os experimentos que utilizaram o tratamento LSA, em geral, adquiriram índices *Silhouette* maiores em relação àqueles que não fizeram uso. Dado esta consideração, realizamos comparativos para cada algoritmo explorado nesta seção a fim apontar esses fatos.
- As representações que no geral mais se saíram bem foram Word2Vec e TFIDF para todos os algoritmos.
- Uma consideração importante a se fazer em relação aos resultados de índice *Silhouette* é a generalização de conhecimento em ambos os códigos. O *Silhouette* indica quanto um dado é coeso ao seu grupo comparado com outros grupos. Em nosso trabalho para as representações utilizadas o índice *Silhouette* alcançou bons valores, contudo para os agrupamentos reais dos dados o *Silhouette* não conseguiu representar de maneira satisfatória as classes.

Por fim, o que houve foi o agrupamento de todos os documentos em apenas dois clusters. Porém, sabe-se de antemão que o número de cluster ideal é cinco para ambos os códigos. Como conclusão, acredita-se que o conhecimento gerado pelos resultados obtidos a partir do índice silhueta não representam os agrupamentos reais para os códigos devido às limitações dos modelos computacionais para representar as classes em questão.

## B. SOM

Foram escolhidos as seguintes variações de parametrização para o uso do algoritmo SOM:

- 1) Taxa de Aprendizado: Por conta da taxa de aprendizado tratar-se de um coeficiente que impacta diretamente no grau de variação dos pesos presentes nos neurônios da rede neural, optamos por três diferentes taxas para comparar o comportamento da rede neural SOM em cada uma delas. Primeiramente, parametrizamos o algoritmo com taxa de aprendizado pequena, mais precisamente com o valor 0.1. Na sequência, realizamos os testes com essa taxa assumindo um valor intermediário de 0.4, e então a utilizamos com o maior valor entre os três, nesse caso 0.7.
- 2) Tipo de decaimento da taxa de aprendizado: Não necessariamente a taxa de aprendizado deve permanecer a mesma durante todo o uso da rede neural. Uma estratégia válida é aplicar um decaimento com o passar das épocas. Existem diferentes formas de aplicar essa redução, para fins comparativos decidiu-se utilizá-la de duas diferentes maneiras. A primeira assume um comportamento linear, ou seja, os valores da taxa de aprendizado decaem de forma uniforme por todo o decorrer do treinamento. A segunda abordagem baseia-se em um comportamento exponencial, o que acarreta em um decaimento mais

expressivo no início do treinamento e, com o passar das épocas, essa variação torna-se mais suavizada. É importante ressaltar que apesar dessas duas abordagens possuírem comportamentos diferentes ambas decaem a taxa de aprendizado do valor inicial até um valor fixo de 0.01.

- 3) Tamanho do látice: O tamanho do látice está diretamente atrelado com a quantidade de neurônios que a rede terá. Não existe um número fixo para a decisão dessa quantidade. Por conta disso, decidimos por tentar três diferentes tamanhos de látice. O menor deles possui uma configuração de dez colunas e dez linhas, totalizando cem neurônios. O caso intermediário é composto por quinze linhas e colunas, portanto duzentos e vinte e cinco neurônios. O maior dos látices utilizados tem uma configuração de vinte linhas e vinte colunas, o que totaliza quatrocentos neurônios.
- 4) Raio de Vizinhança: O raio de vizinhança diz respeito a quantidade de neurônios que movem-se quando um neurônio próximo tem seus valores atualizados, por exemplo caso o raio de vizinhança seja igual a dois, quando um neurônio é realocado no espaço vetorial os outros dois neurônios mais próximos em todos os sentidos no espaço matricial também são movidos. Utilizamos dois diferentes raios, sendo que em ambos os casos, esse valor decaiu linearmente ao longo das eras até alcançar o valor de 1. O primeiro valor de raio usado foi dois. O segundo valor é o menor valor entre o número de linhas e de colunas do látice dividido por dois.
- 5) Tipos de Vizinhança: Dois diferentes tipos de vizinhança foram escolhidos para os devidos testes. A vizinhança Gaussiana e a *Bubble*. A característica que difere ambas está no grau de propagação da atualização de valores entre os neurônios que encontram-se no raio de vizinhança. No caso da *Bubble* a variação de valores que um neurônio receber será propagada inteiramente para os demais neurônios que estão no seu raio de vizinhança, o que não ocorre com a Gaussiana. Nesta a propagação da atualização de valores assume um comportamento diferente, quanto mais próximos no espaço matricial os neurônios vizinhos estão maior será a variação que eles irão receber. Dessa forma vizinhos dentro do raio de vizinhança, porém distantes, oferecerão uma menor mudança.

O restante da parametrização do algoritmo SOM manteve-se constante para todos os experimentos e foi realizada da seguinte maneira:

- o lattice possui geometria planar;
- a grade do lattice é retangular;
- o número de épocas é fixo e igual a 1000;
- a inicialização do algoritmo é aleatória;
- a distância utilizada no Kmeans é a euclidiana.

A clusterização dos documentos utilizando a algoritmo SOM foi realizada aplicando o algoritmo *K-means* nos neurônios do SOM. A partir disso, cada dado do conjunto

de dados foi associado ao cluster de seu respectivo BMU. A clusterização dos neurônios utilizou o *K-means* implementado no *scikit* [4] por conta da fácil integração com o *somoclu* que este possuía. O cálculo do silhueta para o agrupamento SOM utilizou a implementação do grupo.

*a) Resultados SOM e melhores parâmetros:* Os melhores resultados de acordo com o índice silhueta e suas parametrizações são mostradas abaixo:

Tabela VIII

PARAMETRIZAÇÕES DO SOM PARA OS MELHORES RESUTADOS BBC

Sil	LSA	dist	ncluster	proc	representação
0.334	True	Euclidiana	2	Normal	TF
0.263	False	Euclidiana	2	Normal	TF
0.259	True	Euclidiana	2	Lemma	TFIDF

Tabela IX

PARAMETRIZAÇÕES DO SOM PARA OS MELHORES RESUTADOS BBC

Taxa apr	tam. grid	Raio viz	Vizinhança	decaimento
0.1	15	2	Gausiana	Exponencial
0.1	15	7	Gausiana	Exponencial
0.4	10	2	Bubble	Exponencial

Tabela X

PARAMETRIZAÇÕES DO SOM PARA OS MELHORES RESUTADOS REUTERS

Sil	LSA	dist	ncluster	proc	representação
0.325	True	Euclidiana	2	Normal	Word2Vec
0.281	True	Euclidiana	2	Normal	Word2Vec
0.277	False	Euclidiana	2	Normal	Word2Vec

Tabela XI

PARAMETRIZAÇÕES DO SOM PARA OS MELHORES RESUTADOS REUTERS

Taxa apr	tam. grid	Raio viz	Vizinhança	decaimento
0.7	10	5	Bubble	Exponencial
0.1	15	7	Bubble	Exponencial
0.1	20	10	Bubble	Exponencial

*b) Análise dos Parâmetros para o Córpus BBC:* O experimento descrito a seguir apresenta a configuração processamento LSA, quantidade de neurônios igual a 100, taxa de aprendizado 0.7, raio de vizinhança 5, decaimento da taxa de aprendizado exponencial e está apresentado na figura 17.

Notamos que de modo semelhante à clusterização via *K-means*, os melhores resultados ocorrem para as representações Word2Vec e TFIDF, e que a faixa de valores *Silhouette* é a mesma tanto para o Kmeans quanto para a clusterização SOM.

O segundo experimento apresentado na figura 18 possui a parametrização processamento LSA, quantidade de neurônios 15x15=225, taxa de aprendizado 0.4, raio de vizinhança 2 e decaimento exponencial da taxa de aprendizado. Ele ilustra um fenômeno que encontramos em nossos resultados: apesar de encontrarmos em geral os melhores resultados para as representações Word2Vec e TFIDF, a variação dos parâmetros do SOM para o BBC proporcionou resultados que

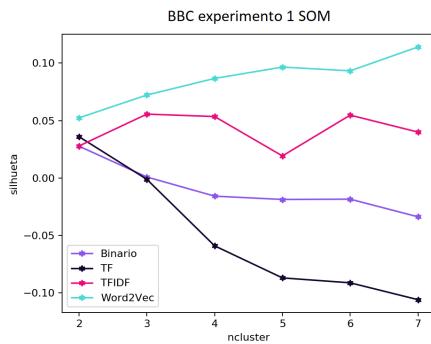


Figura 17.

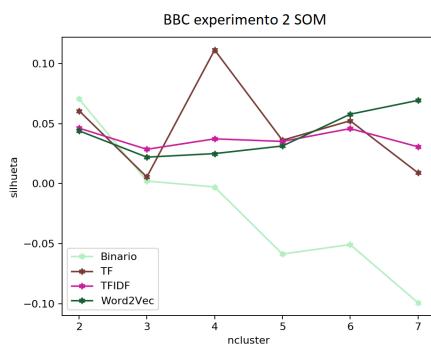


Figura 18.

não seguiam um padrão em específico, variando o formato das curvas de configuração para configuração.

#### c) Análise dos Parâmetros para o Córpus Reuters:

O experimento descrito a seguir possui a parametrização processamento LSA, quantidade de neurônios igual a 225, taxa de aprendizado 0.4, raio de vizinhança 7, decaimento da taxa de aprendizado exponencial e é apresentado na figura 19.

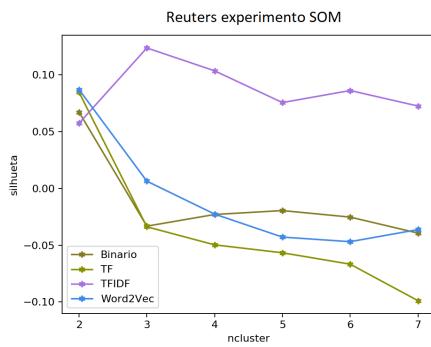


Figura 19.

Notamos que para o Reuters, também as representações Word2Vec e TFIDF possuíram os melhores resultados. Em particular, o TFIDF, assim como mostrado na figura 19, apresenta o melhor resultado entre as representações- padrão

que se aplica a vários outros experimentos também.

*d) Conclusão sobre o SOM:* Os resultados obtidos a partir das execuções do SOM indicam que as representações Word2Vec e TFIDF possuem os melhores índices *Silhouette*, em geral para o número de clusters igual a 2. Contudo, o nosso grupo não conseguiu perceber padrões nas variações das configurações de parâmetros que indicassem que uma variável como o fator de melhoria do modelo. Em vez disso, notamos que combinações de parâmetros geravam os melhores resultados, com alguns exemplos mostrados anteriormente. Apontamos aqui que também houve a questão da generalização de conhecimento, assim como discutida anteriormente.

#### V. Outliers E SUAS IMPLICAÇÕES SOBRE O MODELO

Os conjuntos de dados obtidos a partir de ambos códigos apresentaram um certo nível de desequilíbrio, o que foi evidente a partir de alguns dos resultados de processamento. Para as execuções de todos os números de cluster escolhidos pelo trabalho, foi observado que alguns dos clusters eram compostos por apenas um ou dois dos dados do conjunto, e representavam *outliers*. A estratégia tomada pelo grupo para resolver esse problema foi executar o algoritmo *K-means* e suas variações repetidamente e remover parte dos *outliers* através dos seguintes passos:

- calcular um ponto médio do conjunto de dados;
- calcular as distâncias entre cada dado e o ponto médio;
- remover todos os dados mais distantes do ponto médio equivalente a 3% do conjunto de dados, caso seja do códigos bbc. Caso seja do reuters, remover 7% dos dados.

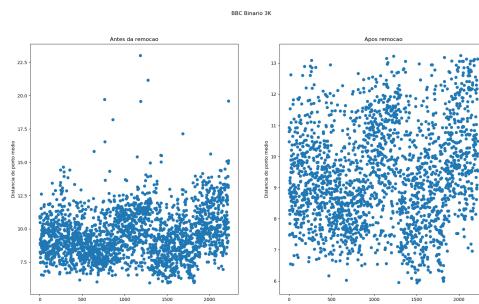


Figura 20.

Os gráficos presentes na figura 20 demonstram um exemplo das disparidades entre as distâncias de dados regulares e de *outliers* para o códigos BBC. Através de análises sobre essas proporções, foi observado que aproximadamente 3% dos dados do códigos BBC representavam *outliers* extremos, enquanto no códigos Reuters esta proporção era de aproximadamente 7%.

Especulamos que possivelmente a remoção de alguns desses *outliers* poderia ser útil para a clusterização, visto que os mesmos poderiam representar um subgrupo insignificante pertencente a uma das categorias dos textos. Por esta razão, o grupo optou por rodar os melhores casos para ambos os dados contendo *outliers* e para os dados com os *outliers* removidos, e assim comparar os resultados.

## VI. PÓS-PROCESSAMENTO

Como o índice *Silhouette* apenas leva em conta o distanciamento dos elementos intra e extra cluster, não se preocupando com o possível conhecimento gerado pela clusterização em questão, observamos uma distribuição insatisfatória dos textos agrupados, apesar dos altos valores desse índice em alguns de nossos testes. Isso ocorreu pois, após testarmos as configurações de nossos algoritmos com amostras de ambos os círpus, o número de clusters que possuía melhor índice *Silhouette* era de apenas dois, quando na realidade os textos possuíam muito mais de duas classes. Sabíamos de antemão que o número de classes era exatamente cinco, o que nos fez realizar testes com esse número de clusters para então observarmos graficamente se cada um dos cinco clusters possuía majoritariamente textos do mesmo assunto.

### A. Silhouette

A medida utilizada para avaliar a qualidade dos agrupamentos foi o índice silhouette. O algoritmo implementado pelo grupo utiliza paralelismo no nível de *threads* para contornar o alto custo computacional que este algoritmo naturalmente exige, por meio da função *Parallel* da biblioteca *joblib* [5]. Outra consideração importante a se fazer sobre a nossa implementação é o reaproveitamento dos cálculos de distância realizados pelo algoritmo para poupar processamento (já que a distância entre mesmos pares de dados são calculado duas vezes) e, assim, otimizar a velocidade do algoritmo permitindo que fosse possível averiguar mais testes, além de aplicar esse mesmo índice no algoritmo *X-means* como decisor para a divisão.

## *B. Nuvem de Palavras*

Como abrir todos os arquivos de cada cluster para checar se a clusterização havia ocorrido de forma a separar consistentemente os textos em suas respectivas classes de acordo com o assunto que cada um se tratava, era uma tarefa inviável, decidimos então utilizar a nuvem de palavras para avaliar qualitativamente a clusterização.

Foi usada para essa tarefa a biblioteca *wordcloud* [8]. Com ela conseguimos plotar as palavras mais frequentes de cada um dos clusters.

Como tínhamos acesso às principais palavras que apareciam em cada cluster poderíamos, por meio dessas, assumir quais eram os principais assuntos que estavam presentes no interior dos textos que compunham cada um desses agrupamentos de uma maneira muito mais eficiente e ainda confiável.

## VII. RESULTADOS

Como mencionado na seção de processamento, obtivemos os resultados para as melhores combinações dos parâmetros observados. A seguir, analisamos qualitativamente o resultado de alguns experimentos que acreditamos serem os mais promissores. A seleção desses experimentos foi baseada nos melhores índices silhueta e no nosso conhecimento prévio de que cada córpus possui cinco classes.

#### A. Resultado 1

Para esse experimento foi utilizado o algoritmo *X-means* no córpus BBC com a parametrização ilustrada na tabela XII.

**Tabela XII**  
**PARAMETRIZAÇÕES DO RESULTADO 1**

Sil	LSA	dist	ncluster	proc	representação
-0.015	True	Cosseno	6	Normal	Binário

Como já havia sido explicitado anteriormente, os resultados do índice *Silhouette* não estão atrelados com uma boa separação semântica dos textos. Por exemplo, nesse resultado tivemos um índice *Silhouette* médio negativo, além de que o maior valor dele dentre todas as trinta iterações foi de apenas 0.038. Porém, ainda assim, ao observamos as nuvens de palavras dos seis clusters que compunham esse resultado, percebemos uma boa qualidade dos agrupamentos para quatro dos clusters.

Os outros dois reuniram palavras que juntas não conseguem determinar de forma clara o tópico abordado pelos documentos.

Relativo à nuvem de palavras da figura 21, conseguimos determinar, a partir das palavras plotadas e das informações de classificação do córpus, que o agrupamento gerado relaciona-se ao tema pré-definido ‘tecnologia’.



Figura 21.

Devido à semântica das palavras que compõem a nuvem de palavras da imagem 22, acredita-se que os documentos reunidos nesse cluster possuem como tema comum ‘economia’ ou ‘economia internacional’.

Na representação da figura 23, podemos observar a prevalência de termos relacionados à política, também um dos temas pré-definidos do córpus BBC. Em destaque, termos relacionados ao ramo executivo de países de origem britânica como *Blair* (Tony Blair, primeiro ministro do Reino Unido de 1997 a 2007), *Howard* (John Howard, primeiro ministro da Austrália de 1996 a 2007), *party* (partido), *labour* (Partido dos Trabalhadores do Reino Unido, a que Tony Blair pertenceu), *election* (eleição), e etc.

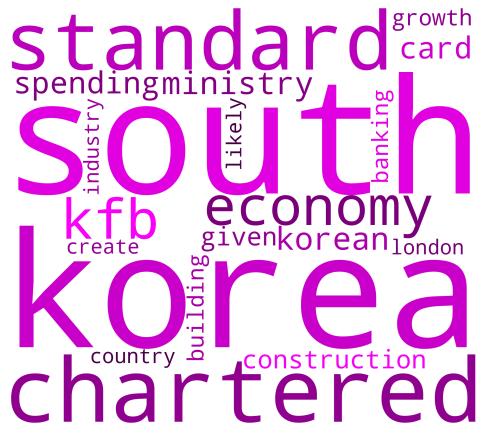


Figura 22.



Figura 23.

Na nuvem de palavras da imagem 24, identifica-se uma grande frequência de palavras relacionadas ao tema ‘entretenimento’, como ‘music’, ‘song’ e ‘Robbie Williams’, apesar da presença de alguns ruídos como ‘wage’, ‘work’ e ‘really’.



Figura 24.

Nos dois clusters das figuras 25 e 26 podemos observar um

alto nível de ruído, especialmente no cluster representado na figura 26. Apesar do cluster da figura 25 apresentar vários termos relacionados ao tema ‘esporte’ do BBC, ele também possui vários termos de ‘entretenimento’ que não formam relações com o tema central. Portanto, não podemos inferir categorias bem definidas para estes clusters.



Figura 25.



Figura 26.

## *B. Resultado 2*

**Tabela XIII**  
**PARAMETRIZAÇÕES DO RESULTADO 2**

Sil	LSA	dist	ncluster	proc	representação
- 0.163	True	Cosseno	5	Normal	Word2Vec

Para esse experimento foi utilizado o algoritmo *K-means* com inicialização ++ no corpus BBC, além de que foram também removidos os *outliers*.

Para o número de clusters igual a cinco, todos os resultados tiveram resultados bastante similares. Na configuração apresentada na tabela XIII, pode-se observar que os clusters representam fielmente os cinco temas do BBC. Apesar da remoção dos *outliers*, não observamos diferenças substanciais

nos resultados, com a exceção da exclusão de alguns ruídos nos primeiros dois clusters.

Na nuvem de palavras da figura 27, pode-se notar que os termos presentes nela referem-se a esportes, mais especificamente a notícias do campeonato europeu de futebol, uma vez que o BBC é um jornal britânico.



Figura 27.

Na imagem 28 observa-se que as palavras deste cluster são termos relacionados a ‘tecnologia’, pois palavras como ‘microsoft’, ‘software’, ‘computer’ se relacionam fortemente à este tópico.



Figura 28.

Percebe-se na figura 29, que os termos são majoritariamente de negócios e economia. Peculiarmente, há a presença de um certo número de palavras relacionadas ao petróleo, inclusive ‘Yukos’, umas das maiores empresas Russas de petróleo.

O cluster representado na figura 30 se refere a entretenimento, mais especificamente a premiações de músicas e filmes.

Observa-se que a nuvem presente na imagem 31 refere-se à política na Inglaterra, com os mesmos termos encontrados no cluster do mesmo tema para o resultado anterior.



Figura 29.



Figura 30.



Figura 31.

### C Resultado 3

Para esse experimento foi utilizado o algoritmo *K-means* com inicialização ++ no córpus BBC, sua parametrização está apresentada na tabela XIV.

Tabela XIV  
PARAMETRIZAÇÕES DO RESULTADO 3

Sil	LSA	dist	ncluster	proc	representação
0.099	True	Euclidian	7	Normal	TFIDF

Em virtude da presença dos nomes de países entre outras palavras no cluster ilustrado pela figura 32, acredita-se que o tema representado por essa nuvem de palavras é ‘política internacional’, cujo não é explicitamente representado pelos rótulos do córpus BBC. Por esse motivo, acredita-se que nesta nuvem de palavras identificou-se um subgrupo originado do tema ‘política’, que não estava presente nos clusters anteriores. Nestes últimos, a grande maioria dos termos estavam relacionados apenas aos países de origem britânica.



Figura 32.

Acreditamos que a nuvem presente na ilustração 33 foi um resultado da intersecção entre os temas ‘tecnologia’ e ‘negócios’. Encontramos palavras relacionadas à telecomunicação, e de teor financeiro. A palavra ‘ebbers’ muito provavelmente se refere a Bernard Ebbers, que contribuiu com a fundação da empresa de telecomunicações conhecida como WorldCom. Em 2005, Bernard Ebbers foi preso por cometer fraudes dentro da própria empresa WorldCom.

As nuvens de palavras presentes na imagens 34 até 38 representam de maneira fiel os rótulos pré estabelecidos pelo córpus. Eles são respectivamente: ‘sport’, ‘tech’, ‘business’, ‘politics’, ‘entertainment’.

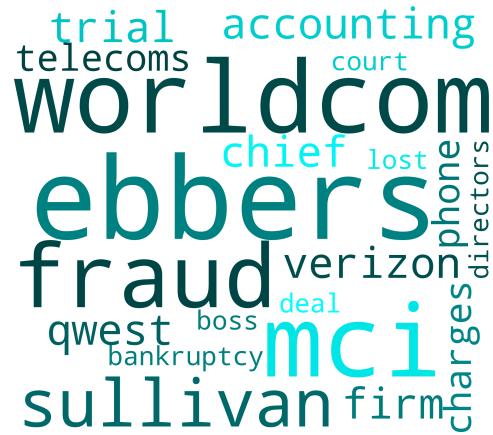


Figura 33.



Figura 34.



Figura 35.

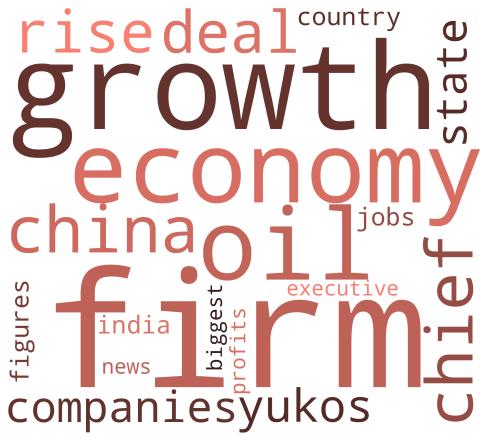


Figura 36.



Figura 37.



Figura 38.

#### D. Resultado 4

Para esse experimento foi utilizado o algoritmo SOM no córpus *Reuters*, a parametrização da rede neural SOM está listada na tabela XV, enquanto a parametrização do algoritmo *K-means*, que foi aplicado para clusterizar os neurônios gerados pelo SOM, está na tabela XVI.

Tabela XV  
PARAMETRIZAÇÕES DO RESULTADO 4

Taxa apr	tam. grid	Raio viz	Vizinhança	decaimento
0.7	10	2	Bubble	Exponencial

Tabela XVI  
PARAMETRIZAÇÕES DO RESULTADO 4

Sil	LSA	dist	ncluster	proc	representação
0.045	True	Euclidiana	2	Lemma	TFIDF

Preliminarmente, realizamos a plotagem da nuvem de palavras para as palavras mais comuns entre todos os textos, porém obteve-se resultados inconclusivos, pois existiam muitas palavras consideradas ruídos como ‘blah’, ‘dlr’, ‘mln’, ‘pct’, como está representado na imagem 39.

No caso ‘blah’ é um ruído que aparece após os começos de cada notícia. ‘Dlr’, ‘mln’ e ‘pct’ significam *dólar*, *million* e *percent*, respectivamente. Essas e outras palavras ruídos foram considerados *stop words*, já que eles apareciam frequentemente em todos ou quase todos os clusteres.

Após a remoção desses ruídos, a tarefa de interpretação semântica da nuvem de palavras tornou-se muito mais factível. A imagem 40 ilustra o mesmo cluster que na imagem 39 porém com essa remoção.



Figura 39.

Porém, mesmo após esse tratamento prévio, ainda assim, não foi possível conseguir uma boa conclusão sobre qual assunto cada um dos dois clusters finais se tratavam, uma vez que ambos ainda apresentavam, palavras que, em sua maioria, se tratavam do mesmo assunto ou que fossem até mesmo iguais, conforme a análise feita das imagens 40 e 41.

Esse comportamento nas clusterizações foi majoritário por quase todo o córpus *Reuters*, sendo assim muito rasa a quantidade de informação passível de interpretação durante a análise dos agrupamentos gerados por todas as diferentes técnicas e parametrizações utilizadas. O grupo acredita que isso tenha ocorrido devido a similaridade dentre os assuntos abordados pelos textos que compunham o córpus *Reuters* inteiramente.

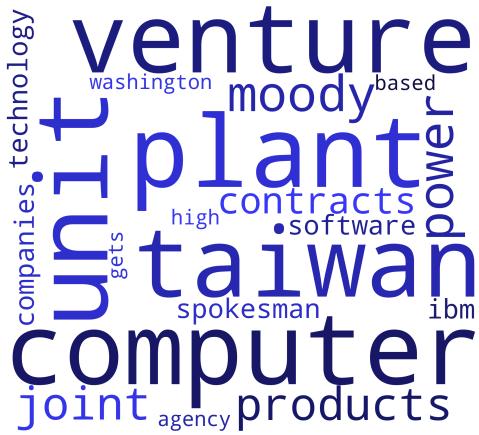


Figura 40.



Figura 41.

## VIII. CONSIDERAÇÕES FINAIS

Ao possuirmos análises robustas dos resultados com uma grande quantidade de variações paramétricas de todos os nossos algoritmos conseguimos convergir nossas opiniões em algumas conclusões.

Devido a discrepância da qualidade do conhecimento obtido no pós-processamento entre ambos os córpuses, percebemos que a tarefa de agrupamento de textos por classes de assuntos é muito mais propícia para o córpus BBC. Isso porque ao analisarmos as nuvens de palavras provenientes desse córpus foi possível obter uma separação de contextos totalmente diferentes presentes em cada agrupamento. Diferentemente do Reuters, cujo no caso não importava a quantidade de clusters que usássemos, sempre a maioria deles apresentava uma grande quantidade de palavras que tratavam-se de assuntos análogos. Assim, concluímos que, com o uso desse córpus, não chegamos em um agrupamento de textos satisfatório.

Observamos também que, comparativamente, os melhores resultados gerados pelos SOM não chegavam a prover tanto conhecimento quanto os melhores do algoritmo *K-means* e suas variações. Por conta disso, chegamos a conclusão que, para a realização da tarefa proposta, o uso da rede neural SOM

não se sai tão bem quanto o uso das técnicas de agrupamento utilizadas ao decorrer do trabalho.

Outro ponto importante que percebemos foi que após a retirada dos *outliers* dos córpuses, ao testarmos novamente nossos algoritmos com a mesma parametrização, os resultados obtidos tiveram uma melhora quase que imperceptível. Contrariando o que imaginávamos que ocorreia. Logo, concluímos com isso que o impacto dos *outliers* em nossos córpuses utilizando as técnicas que selecionamos para o processamento é praticamente ínfimo.

Foi percebido também que as representações Word2Vec e TFIDF ao serem plotadas no inicio do trabalho possuam um grau de espalhamento maior do que as demais representações, o que poderia facilitar na tarefa de agrupamento. Acreditamos que isso acarretou nos resultados superiores que ambas tiveram quando comparadas às representações TF e binária.

Assim como a remoção de *outliers*, o uso da técnica de Lematização também não trouxe melhorias suficientemente expressivas aos resultados finais. Isso pode ter sido constatado devido a baixa variação do índice *Silhouette* quando comparado com testes que seguiam a mesma parametrização, porém sem o uso da Lematização.

Ao observarmos as nuvens de palavras que foram geradas a partir dos clusters que não possuiam índices *Silhouette* razoavelmente altos, sendo estes passíveis de melhores extrações de informação que outros que apresentavam um índice maior, acabamos por concluir que o conhecimento oriundo dos resultados obtidos a partir dos melhores índices *Silhouette* não representam os agrupamentos reais para os córpuses devido às restrições dos modelos computacionais para representar as classes em estudo. Por esse motivo comprehende-se que o índice *Silhouette* não pode ser assumido como o único avaliador a fim de julgar as diferentes parametrizações dos algoritmos usados. Contudo, conseguimos alcançar bons resultados tendo em vista o conhecimento a priori dos rótulos para o córpus BBC. Em relação ao córpus Reuters, não conseguimos clusterizá-lo de modo satisfatório, mesmo considerando informações prévias sobre esse conjunto de textos.

## REFERÊNCIAS

- [1] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70. [Online]. Available: <https://doi.org/10.3115/1118108.1118117>
- [2] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [3] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science and Engg.*, vol. 13, no. 2, pp. 22–30, Mar. 2011. [Online]. Available: <https://doi.org/10.1109/MCSE.2011.37>
- [4] 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*, vol. 12, pp. 2825–2830, 2011.
- [5] Joblib: running python functions as pipeline jobs. [Online]. Available: <https://pythonhosted.org/joblib/>
- [6] Somoclu. [Online]. Available: <https://github.com/peterwittek/somoclu>

- [7] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.
- [8] word cloud. [Online]. Available: <https://github.com/amueller/word-cloud>
- [9] Bbc. [Online]. Available: <http://mlg.ucd.ie/howmanytopics/index.html>
- [10] Reuters. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection>
- [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: <http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781>
- [12] M. Á. Á. Carmona, A. P. López-Monroy, M. Montes-y-Gómez, L. V. Pineda, and H. J. Escalante, "Inaoe's participation at pan'15: Author profiling task," in *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*, 2015. [Online]. Available: <http://ceur-ws.org/Vol-1391/122-CR.pdf>
- [13] Dive into nltk, part iv: Stemming and lemmatization. [Online]. Available: <https://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization>
- [14] Google news embeddings. [Online]. Available: <https://code.google.com/archive/p/word2vec/>
- [15] Wikipedia contributors, "T-distributed stochastic neighbor embedding — Wikipedia, the free encyclopedia," [https://en.wikipedia.org/w/index.php?title=T-distributed\\_stochastic\\_neighbor\\_embedding&oldid=843641154](https://en.wikipedia.org/w/index.php?title=T-distributed_stochastic_neighbor_embedding&oldid=843641154), 2018, [Online; accessed 4-June-2018].
- [16] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 727–734. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645529.657808>