Análise de Sentimentos de Conteúdos Textuais de Redes Sociais Por Meio de Modelos de Compressão de Dados

Jurandir J. D. Silva[1], **Rogerio Salvini**[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG) – Goiânia, GO – Brazil

{jurandirsilva,rogeriosalvini}@inf.ufg.br

Resumo. *O conjunto das técnicas que são exploradas no tratamento de opiniões é abordado pela área de pesquisa da Análise de Sentimentos (AS), que combina conceitos de diversas áreas como, Inteligência Artificial, Reconhecimento de Padrões, Análise Textual etc. Por outro lado, técnicas baseadas em compressão de dados podem ser úteis para achar padrões em dados não estruturados, como textos com opiniões encontrados na internet. Neste trabalho foi testado o método DAMICORE, que utiliza estas técnicas, para verificar sua eficácia no problema de AS. Os resultados ficaram aquém dos esperados, entretanto abrindo novas oportunidades de pesquisa na área.*

1. Introdução

Atualmente, sistemas web como blogs, redes sociais, sites de compras etc., permitem cada vez mais a interação dos usuários, e a utilização de textos para comunicação e/ou expressão de opinião tornou-se relativamente comum. De forma natural, o conteúdo destes textos se tornou objeto de atenção para os maiores interessados no "ponto de vista" de seu público: as empresas. De acordo com Giachanou [Giachanou and Crestani 2016], a informação gerada pelos usuários da internet é uma boa fonte de opiniões e pode ser valiosa para uma variedade de aplicações que requerem uma compreensão da opinião pública sobre um conceito. Um exemplo típico que ilustra a importância da opinião pública refere-se às empresas que podem capturar as opiniões dos clientes sobre seus produtos ou seus competidores. Por outro lado, a organização desses dados para se inferir correlações em geral não tem sido um trabalho trivial. A dificuldade em se automatizar esse tipo de processamento ocorre, principalmente, porque a quantidade de formas diferentes de se expressar uma opinião, ou mesmo um sentimento, é relativamente grande.

O conjunto das técnicas que são exploradas no tratamento de opiniões é abordado pela área de pesquisa de Análise de Sentimento, que combina conceitos de diversas áreas como, Inteligência Artificial, Reconhecimento de Padrões, Aprendizado de Máquina, Processamento de Linguagem Natural e Análise Textual; objetivando analisar fragmentos textuais e determinar a atitude, emoção, opinião, avaliação ou sentimento do usuário com relação a algum tópico ou entidade [B. Liu 2012]. Dentre as técnicas de

aprendizado de máquina que vêm sendo empregadas na Análise de Sentimento estão: *Support Vector Machines* (SVM), *Naïve Bayes* (NB) e *Multinomial Naïve Bayes* (MNB) [Giachanou and Crestani 2016]. Além dessas técnicas, algoritmos evolutivos (*Evolutionary Algorithms*) [Dufourq and Bassett 2017], e o algoritmo de propagação (*Propagation Algorithm*) [Che et al. 2015], ambos utilizando compressão de dados, já foram empregados para análise de sentimento, através de uma abordagem supervisionada. No entanto, abordagens não supervisionadas, como a Análise de Agrupamentos (ou *Clustering*), ainda não estão muito exploradas nessa área.

A Análise de Agrupamentos é o conjunto de técnicas computacionais cujo propósito consiste em separar objetos em grupos, baseando-se nas características que estes objetos possuem. O agrupamento torna-se um mecanismo útil quando se tem a necessidade de analisar dados de diferentes tipos e tamanhos, assim como é possível extrair características não conhecidas anteriormente de um determinado grupo.

Modelos para análise de agrupamentos utilizando compressão de dados já se mostraram eficazes quando utilizados no agrupamento de textos [Sanches et al. 2011]. Neste contexto, os compressores geram arquivos compactados que são usados como padrões para determinar grupos semelhantes, formando *clusters* de objetos textuais.

O objetivo deste trabalho é aplicar e verificar se estas técnicas também podem ser eficazes no problema de análise de sentimento agrupando mensagens provenientes de websites como Twitter (*tweets*) e Amazon (comentários sobre produtos) em função dos sentimentos positivo ou negativo expressos através das mensagens dos usuários. Para isto, será utilizado um método chamado DAMICORE (*DAta MIning of COde REpositories*) que será apresentado em maiores detalhes na Seção 2 a seguir.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta o método DAMICORE para geração de agrupamentos baseado em compressão de dados; a Seção 3 aborda a metodologia empregada nos experimentos deste trabalho; a Seção 4 mostra os resultados alcançados; a Seção 5 faz uma discussão sobre estes resultados e levanta algumas questões sobre sua utilização para o problema abordado; e finalmente a Seção 6 faz uma conclusão deste estudo e apresenta alguns trabalhos futuros.

2. DAMICORE (*DAta MIning of COde REpositories*)

O DAMICORE[1] é um método proposto por Soares [Soares and Delbem] para análise de agrupamentos (*Clustering*) que se baseia em procurar similaridades nos dados utilizando compactação de dados, que são técnicas utilizadas para diminuir o volume da informação [Sanches et al. 2011].

A geração dos agrupamentos por meio do DAMICORE segue a combinação dos

[1]Disponível em: https://github.com/sidgleyandrade/damicore-python

seguintes algoritmos: *Normalized Compression Distance* (NCD), proveniente da Teoria da Informação [Cilibrasi and Vitányi 2005], o *Neighbor Joining* (NJ), proveniente da Filogenética [Felsenstein J. 2003], e o *Fast Algorithm* (FA), proveniente de Redes Complexas [Newman 2004].

A NCD (do inglês, Distância por Compressão Normalizada) é uma métrica que possibilita quantificar a semelhança entre diferentes dados a partir da quantidade de informação do arquivo compactado (tamanho do arquivo). Esta abordagem não requer qualquer conhecimento específico do domínio de aplicação. De acordo com Cilibrasi [Cilibrasi and Vitányi 2005], a NCD é uma métrica universal e robusta que tem sido aplicada com sucesso em áreas como a genética, literatura, música e astronomia. A NCD é dada pela seguinte equação:

$$NCD(X,Y) = \frac{C(XY) - min\{C(X),C(Y)\}}{max\{C(x),C(y)\}} \quad (1)$$

onde, $C(XY)$ é o tamanho dos arquivos X e Y concatenados e, em seguida, compactados; $C(X)$ e $C(Y)$ são os tamanhos dos arquivos X e Y compactados, respectivamente.

A NCD é usada no DAMICORE para gerar matrizes de distância para um conjunto de variáveis, possibilitando verificar níveis de semelhança entre elas e propriedades em comum desses códigos que podem ser explorados.

O algoritmo NJ possibilita agrupar arquivos, a partir de sua semelhança, formando clusters de arquivos que compartilham alguma característica em comum. O NJ constrói uma árvore a partir de uma matriz de distâncias evolutivas, adaptando o critério de evolução mínima, em que o algoritmo tenta minimizar a soma dos tamanhos de todos os nós da árvore. A ideia central da técnica é identificar pares de objetos mais próximos.

O *Fast Algorithm* (FA), proposto em [Newman 2004], é um algoritmo de Detecção de Comunidades, cujo objetivo é encontrar a estrutura que maximize a fração de arestas que conectam vértices de uma mesma comunidade [Crocomo 2012].

O DAMICORE funciona seguindo uma sequência de etapas abaixo:

- Etapa 1: O conjunto de objetos é reorganizado de forma a compor uma amostra;
- Etapa 2: É gerada uma matriz que relaciona cada objeto com os demais através da NCD. Nesta etapa, a NCD necessita de um compressor, que é indicado pelo usuário;
- Etapa 3: É gerada uma árvore filogenética usando a heurística NJ a partir da matriz de distâncias construída na Etapa 2. O DAMICORE faz uso da técnica de árvores aditivas, sem raiz, em que a distância entre dois objetos é dada pela soma dos

ramos que os unem;

- Etapa 4: O DAMICORE utiliza a saída do algoritmo NJ, que será um arquivo no formato Newick [Felsenstein J. 2003], para formar uma Matriz de Adjacências.
- Etapa 5: É determinado um Particionamento Final a partir da Matriz de Adjacências da Etapa 4. Para isso, é utilizado o algoritmo FA;

3. METODOLOGIA

3.1. Bases de Dados

As bases de dados utilizadas nos resultados expostos neste trabalho foram as bases *Sanders*[2] e a *Sentiment Labelled Sentences Data Set*[3], ambas compostas de dados reais e disponíveis publicamente.

3.1.1. Sanders

A base *Sanders* é composta por mensagens provenientes do Twitter (*tweets*), e foi escolhida por já ser utilizada em diversos artigos, o que possibilita uma melhor comparação com os resultados aqui obtidos.

Esta é formada por mensagem obtidas de forma manual por especialistas, e foram coletadas a partir de quatro termos de buscas: *@apple*, *#google*, *#microsoft* e *#twitter*. A Tabela 1 apresenta o número de *tweets* de cada tópico e classe:

Tabela 1. Especificação da base de dados *Sanders*

Tópico	Classe Positiva	Classe Neutra	Classe Negativa	Termo de busca no Twitter
Apple	191	581	377	*@apple*
Google	218	604	61	*#google*
Microsoft	93	671	138	*#microsoft*
Twitter	68	647	78	*#twitter*

Por se tratar de um estudo que tem por objetivo verificar se o DAMICORE pode ser eficaz no problema de análise de sentimento, e também visando uma melhor avaliação dos resultados, os exemplos da classe neutra da *Sanders* foram ignorados, permanecendo apenas as classes positiva e negativa.

3.1.2. Sentiment Labelled Sentences Data Set (SLSDS)

A base *Sentiment Labelled Sentences Data Set* é formada por comentários/avaliações de usuário nos *websites*: *amazon.com*, *imdb.com* e *yelp.com*; sobre produtos, filmes e restaurantes, respectivamente [Kotzias et al. 2015].

[2]Disponível em http://www.sananalytics.com

[3]Disponível em https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

Algumas diferenças desta base com relação à base *Sanders* são:

- Os exemplos não são sobre termos/temas específicos;
- Os exemplos foram selecionados de outras bases maiores, de forma que esta se manteve balanceada (mesma quantidade de exemplos de cada classe);
- Das bases maiores, foram selecionados apenas as sentenças com conotação clara de "positivo" (indicados na base por 1) ou "negativo" (indicados na base por 0);

A Tabela 2 a seguir apresenta o número de sentenças de cada *website* e classe:

Tabela 2. Especificação da base de dados *Sentiment Labelled Sentences Data Set*

Website	Classe Positiva	Classe Negativa
amazon.com	500	500
imdb.com	500	500
yelp.com	500	500

3.2. Pré-processamento

Para a realização deste trabalho foi desenvolvido um pacote de *scripts*, chamado *pyDataPreparation*[4], desenvolvido na linguagem Python 3, que é responsável por toda a preparação e o pré-processamento dos dados.

O pré-processamento foi realizado em etapas, onde a cada experimento realizado, quando feito o pré-processamento, foi adicionado pelo menos um item de pré-processamento dos dados. Os itens de pré-processamento foram os seguintes:

1. Passagem de todas as letras maiúsculas para minúsculas;
2. Remoção de todos os símbolos não alfanuméricos (que não são letras ou números), mantendo os espaços;
3. Remoção das *stopwords*, utilizando o conjunto de palavras presentes na *corpora stopwords* da biblioteca NLTK (*Natural Language Toolkit*)
4. Substituição de cada palavra, quando possível, por um sinônimo "mais relevante". Por exemplo, caso as palavras "*happy*" e "*glad*" apareçam nas mensagens, mas a palavra "*happy*" aparece com mais frequência, todas as ocorrências da palavra "*glad*" são substituídas pela palavra "*happy*";

É importante frisar que o conteúdo das sentenças após o pré-processamento é compostos apenas pelas mensagens propriamente ditas, ou seja, a classe (ou algo que a indique) não é inserida nos arquivos finais, com exceção do nome apenas para identificação.

[4]Disponível em: https://github.com/jurandirjdsilva/pyDataPreparation

3.3. Compressor de Dados

Como abordado por Cilibrasi em [Cilibrasi and Vitányi 2005], diversos compressores de dados podem ser utilizados no cálculo da NCD. Neste trabalho, os experimentos foram realizados utilizando o compressor GZIP [5], que utiliza o algoritmo de compressão *Lempel–Ziv*. Em [Cilibrasi and Vitányi 2005], o GZIP foi escolhido para comprimir sequências de textos cujo tamanho não excede a janela deslizante do compressor (32 *kilobytes*), que é justamente o caso dos experimentos deste trabalho, em que os maiores arquivos têm tamanhos que se aproximam de 200 *bytes*.

4. RESULTADOS

Diversos experimentos, em diferentes situações, foram executados. No entanto, para fim de uma melhor discussão posteriormente, os resultados aqui relatados correspondem a apenas os resultados mais relevantes alcançados. Por resultados mais relvantes entende-se os que tiveram mais impacto, seja sob os resultados esperados, seja na comparação com outros trabalhos.

Os resultados apresentados nesta seção representam as filogenias resultantes da execução do algoritmo NJ. Tais filogenias são exibidas neste trabalho na forma de Cladogramas Diagonais, cuja interpretação é como segue: O NJ identifica pares de objetos mais próximos, chamados de nós vizinhos, que são conectados por um nó mais interno formando uma subárvore bifurcada [Valdivia 2007]. A relação de vizinhança é ilustrada nos gráficos aqui apresentados na forma de linhas que conectam as folhas (os objetos dos experimentos) formando nós, e linhas que unem esses nós a outas folhas ou outros nós; de maneira que quanto menos nós houver entre dois objetos mais próximos (semelhantes) estes são.

O cladograma diagonal é semelhante a um dendrograma, bastante utilizado na análise de agrupamentos, estando a diferença na representação das distâncias, onde no dendrograma estas ficam explícitas no tamanho das arestas enquanto que no cladograma as distâncias ficam representadas apenas nos níveis. Devido à quantidade de exemplos nas bases utilizadas, nos cladogramas gerados não é possível ler claramente os rótulos dos exemplos (devido à restrição de espaço). Assim, para facilitar a interpretação dos resultados, optamos por representar os exemplos por diferentes cores conforme a sua classe. Os rótulos dos exemplos foram coloridas manualmente a fim de facilitar a identificação dos subgrupos gerados pelo algoritmo. As correspondências entre cor e classe, no Experimento 1, encontra-se na legenda da figura. Já nos Experimentos 2 ao 8, as sentenças que expressam sentimento positivo estão destacados com a cor verde e negativo com a cor vermelha.

[5] Disponível em: https://www.gnu.org/software/gzip/

A fim de possibilitar uma melhor avaliação dos resultados, o primeiro experimento abordado nesta sessão foi realizado utilizando-se a base de dados que acompanha o DAMICORE, e servirá de *baseline* para a comparação com os resultados obtidos nas bases de dados de análise de sentimentos utilizadas neste trabalho.

4.1. Experimento 1: Base de dados de textos em diferentes idiomas

A base de dados utilizada neste experimento é composta por arquivos textuais de cinco idiomas diferentes, sendo eles: alemão, francês, inglês, italiano e português; contendo ao todo 150 exemplos, 30 de cada classe/idioma.

Cada texto é armazenado em um arquivo de forma que as três primeiras letras desses arquivos representam o idioma do texto, da seguinte maneira: “GER” para o idioma alemão; “FRA” para o idioma francês; “ENG” para o idioma inglês; “ITA” para o idioma italiano; e “POR” para o idioma português.

O cladograma diagonal apresentado na Figura 1, obtido a partir do arquivo *newick* que é gerado, resume o resultado do DAMICORE, exibindo os *cluesters* na forma de clados. Pode-se notar nesta figura que os agrupamentos dos textos por idioma ficaram perfeitos.

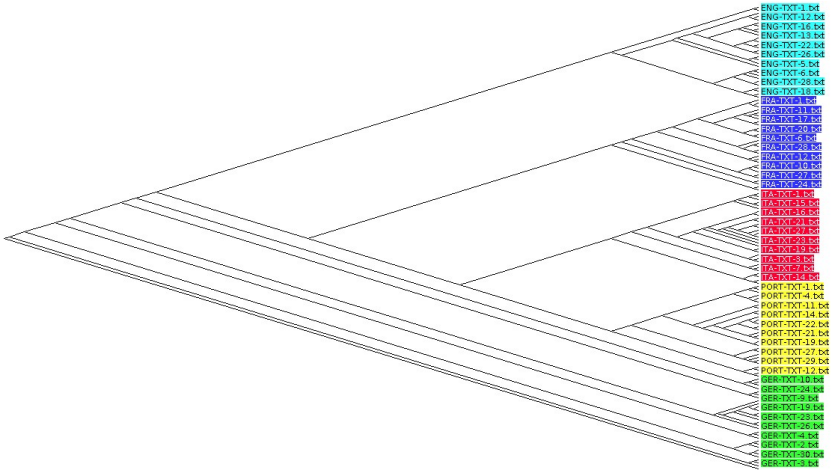

Figura 1. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre a base de dados de textos em diferentes idiomas. Na figura, textos do idioma inglês estão destacados com a cor azul claro, francês na cor azul escuro, italiano na cor vermelho, português na cor amarelo e alemão na cor verde.

4.2. Experimento 2: Base Sanders completa sem pré-processamento

Tabela 3. Descrição do Experimento 2

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamento
Sanders completa	1219	565	654	Sem pré-processamento

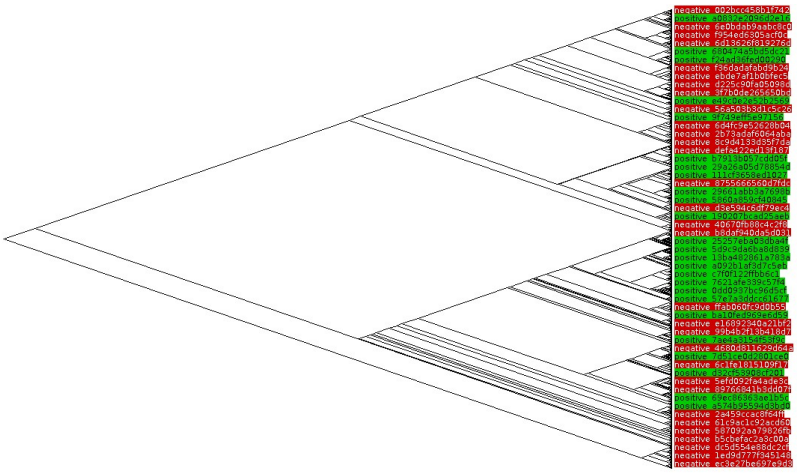

Figura 2. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre a base de dados Sanders completa sem nenhum pré-processamento.

4.3. Experimento 3: Base Sanders completa com remoção de símbolos e *stopwords*

Tabela 4. Descrição do Experimento 3

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamento
Sanders	1217	563	654	Letras minúsculas; Remoção de símbolos não alfa-numéricos; Remoção de *stopwords*

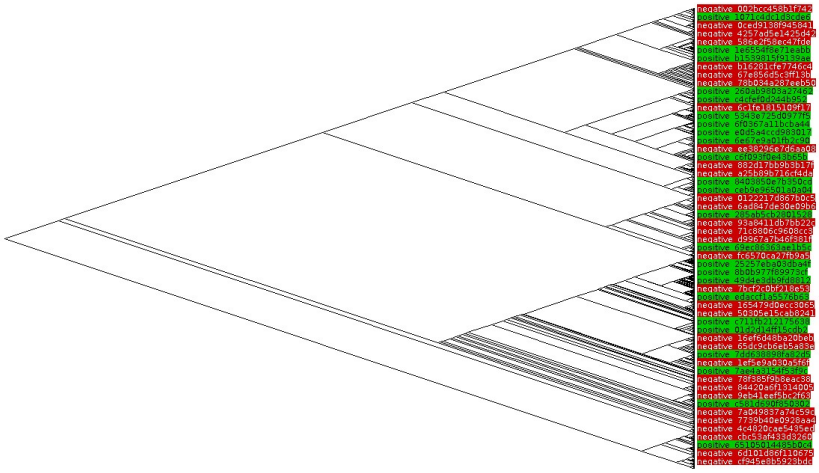

Figura 3. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre a base Sanders completa e pré-processamento descrito na Tabela 4.

4.4. Experimento 4: Base Sanders completa com substituição de sinônimos

Tabela 5. Descrição do Experimento 4

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamento
Sanders	1217	563	654	Letras minúsculas; Remoção de símbolos ; Remoção de *stopwords*; Substituição de palavras por sinônimos

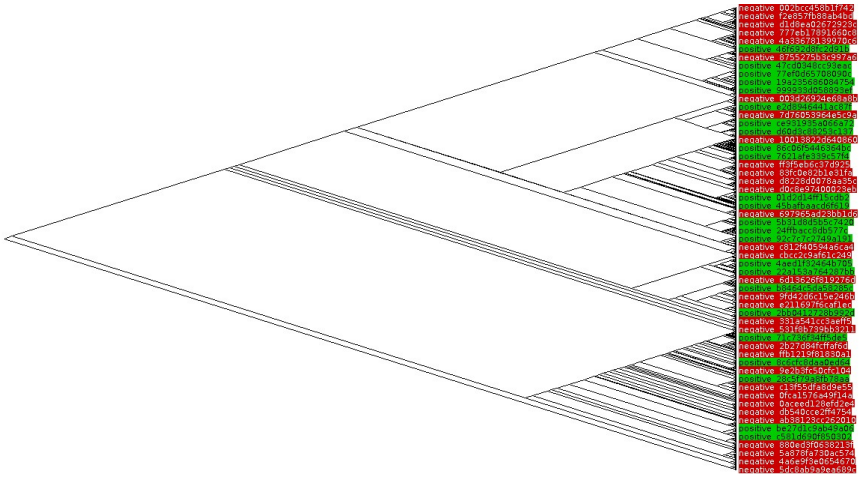

Figura 4. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre a base Sanders completa e pré-processamento descrito na Tabela 5.

4.5. Experimento 5: Subconjunto referente ao termo #apple da base Sanders sem pré-processamento

Tabela 6. Descrição do Experimento 5

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamento
Sanders: apple	568	191	377	Sem pré-processamento

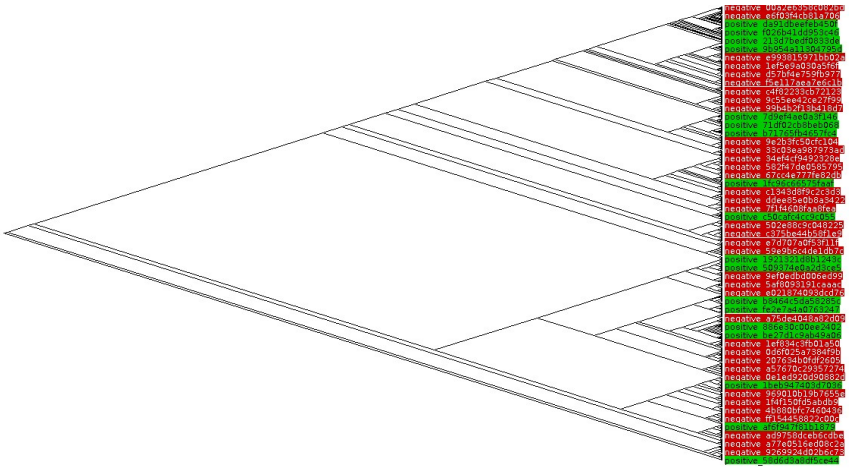

Figura 5. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre o subconjunto da base Sanders referente ao termo #apple e sem pré-processamento.

4.6. Experimento 6: Subconjunto balanceado referente ao termo #apple da base Sanders sem pré-processamento

Tabela 7. Descrição do Experimento 6

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamentoo
Sanders: apple	380 (selec. aleat.)	190	190	Sem pré-processamento

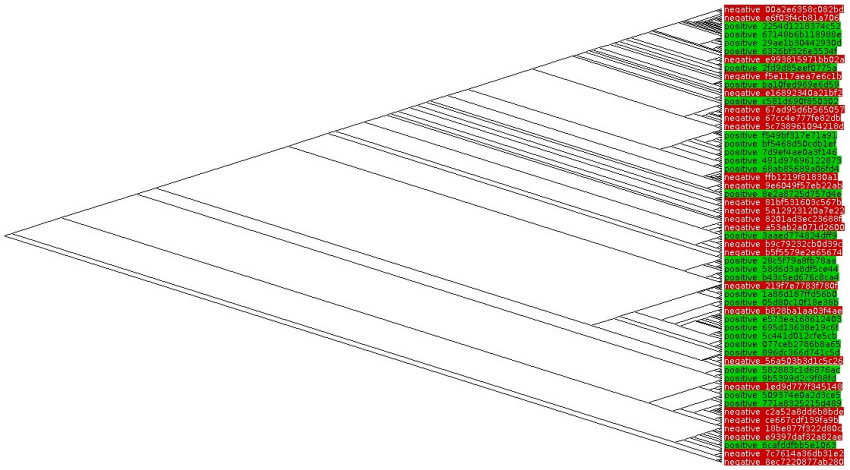

Figura 6. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre o subconjunto da base Sanders referente ao termo #apple e sem pré-processamento.

4.7. Experimento 7: Subconjunto balanceado referente ao termo #apple da base Sanders com remoção de símbolos e *stopwords*

Tabela 8. Descrição do Experimento 7

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamento
Sanders: apple	380 (selec. aleat.)	190	190	Letras minúsculas; Remoção de símbolos não alfanuméricos; Remoção de *stopwords*

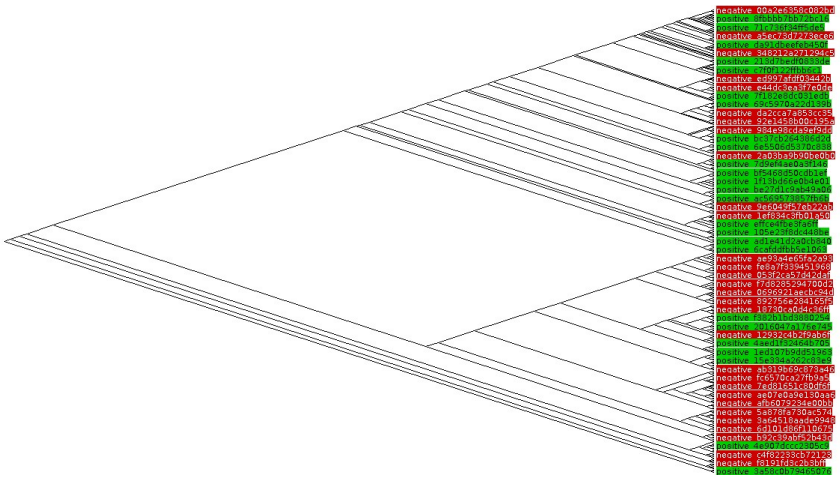

Figura 7. Cladograma diagonal mostrando o resultado do DAMICORE executado sobre o subconjunto da base Sanders referente ao termo #apple e pré-processamento descrito da Tabela 8.

4.8. Experimento 8: Subconjunto balanceado proveniente do *website* Amazon da base *Sentiment Labelled Sentences Data Set* com remoção de símbolos e *stopwords*

Tabela 9. Descrição do Experimento 8

Base de Dados	#Exemplos	#Pos.	#Neg.	Pré-processamento
SLSDS: Amazon	200 (selec. aleat.)	100	100	Letras minúsculas; Remoção de símbolos não alfanuméricos; Remoção de *stopwords*

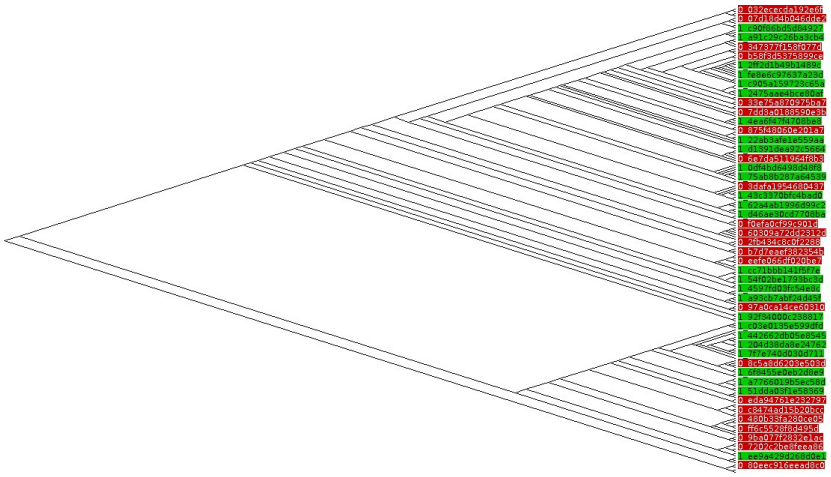

Figura 8. Cladograma diagonal mostrando o resultado do DAMICORE executando sobre o subconjunto da base Sentiment Labelled Sentences Data Set provenientes do *website* Amazon e pré-processamento descrito da Tabela 9

5. DISCUSSÃO

O Experimento 1 usou o DAMICORE para achar agrupamentos em uma base de textos em diferentes idiomas. Esta base é obtida junto com o código do DAMICORE e foi utilizada como *baseline* deste trabalho, ou seja, o cladograma do resultado obtido nesta base de dados é utilizado como modelo do que se esperava dos resultados nas bases escolhidas para a análise de sentimentos. Foram realizadas verificações minuciosas, tanto no cladograma em tamanho original, quanto no arquivo com os *clusters* gerado pelo DAMICORE, e constatou-se que todos os *clusters* gerados continham textos de apenas 1 idioma. Ou seja, o DAMICORE se mostrou realmente eficaz na tarefa de agrupamento de textos por idiomas.

Nos experimentos 2, 5 e 6, o DAMICORE foi executado sobre as bases de dados voltadas para análise de sentimentos sem nenhum pré-processamento. Conforme abordado em [Ziegelmayer and Schrader 2012], a classificação de textos baseada em compactação praticamente não requer pré-processamento, pois é capaz de capturar *non-words* (grupos de letras que se parecem com determinadas palavras) e *metawords* (carac-

terísticas que abrangem mais de uma palavra). No entanto, nestes experimentos realizados, o DAMICORE não foi capaz de agrupar os exemplos segundo sua classe, como é possível verificar através dos cladogramas em cada subseção, em que os exemplos das duas classes (destacados pelas cores vermelho e verde) ficam "misturados" entre si. Em um resultado positivo, seria possível observar dois grandes agrupamentos, sendo cada um formado, em sua maioria, por exemplos de uma mesma classe.

Nos experimentos 3, 4, 7 e 8, os dados passaram por pré-processamento a fim auxiliar o método. No entanto, o DAMICORE também não obteve sucesso nesses casos.

No experimento 4, um dos pré-processamentos adotados nas sentenças foi a substituição de cada palavra, quando possível, por um sinônimo mais "relevante" (com maior probabilidade de ocorrência nas demais sentenças). Essa medida foi adotada com o objetivo de aumentar a similaridade entre as sentenças (quanto maior a similaridade, menor o valor da NCD), sem alterar o significado destas. Porém, nem mesmo isso foi suficiente para melhorar a eficiência do DAMICORE.

É interessante notar que, em alguns experimentos, como o 2 e o 7, ocorrem alguns subagrupamento com uma quantidade relevante de exemplos de mesma classe. Diante disso, poderíamos concluir que o DAMICORE funciona parcialmente na análise de sentimentos. No entanto, a ocorrência desses subagrupamentos ocorre de forma desordenada, não sendo possível prever quando e como esses subagrupamentos surgirão.

Como os resultados não foram os esperados, estudos prévios foram realizados e, pelo menos, duas razões para a ineficiência do método foram descobertas:

- De acordo com Cilibrasi [Cilibrasi and Vitányi 2005], certas características dominantes que regem a similaridade são automaticamente descobertas pelo NCD. Além disso, essas característica podem não ser explicitamente conhecidas para nós. Isso significa que o NCD pode agrupar dados de acordo com características que não são de interesse do problema em estudo e, desta forma, o resultado final (os *clusters*) não necessariamente estará incorreto.
- A NCD, apresentada em [Cilibrasi and Vitányi 2005], é definida como uma medida de similaridade que expressa uma distância entre dois objetos quaisquer por meio da compressão dos dados. No entanto, alguns testes prévios foram realizados e constatou-se que, em alguns casos, a distância entre dois objetos ($NCD(A, B)$, para $A \neq B$) é diferente ao se inverter a ordem dos objetos ($NCD(B, A)$). Isso pode ser um motivo real da ineficiência da NCD para o agrupamento de objetos, uma vez que, por se tratar de uma medida de similaridade, as distâncias $NCD(A, B)$ e $NCD(B, A)$ tinham que ser exatamente as mesmas. Esse fato coloca em risco a própria validade desta medida como uma métrica de distância.

6. CONCLUSÃO

Devido à eficácia do método DAMICORE no agrupamento de textos por idiomas, no presente trabalho, este mesmo método foi aplicado no problema de análise de sentimentos, para verificar se técnicas baseadas em compressão podem ser também eficazes agrupando mensagens de texto em função do sentimento expresso através destas.

Constatou-se porém, por meio de vários experimentos, e utilizando diversos tipos de pré-processamento dos dados a fim de ajudar o método, que o DAMICORE não foi capaz de formar agrupamentos de mensagens que tenham como característica em comum o sentimento (positivo ou negativo) expresso.

Dentre as razões que podem ter contribuído para este insucesso do DAMICORE está a possibilidade do método estar formando grupos a partir de características que não estão relacionadas às palavras referentes aos sentimentos nas mensagens. Além disso, a métrica utilizada para fazer o agrupamento dos objetos, a NCD, não é simétrica ($NCD(A, B) \neq NCD(B, A)$) o que interfere diretamente no agrupamento dos objetos.

Como trabalhos futuros, serão realizados novos estudos com o objetivo de entender melhor as limitações do DAMICORE, bem como realizar modificações no método a fim de aperfeiçoá-lo. Dentre as modificações possíveis, será considerada a implementação de um novo compressor de dados, além de possíveis modificações na métrica NCD.

Referências

B. Liu (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Che, W., Zhao, Y., Guo, H., Su, Z., and Liu, T. (2015). Sentence compression for aspect-based sentiment analysis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(12):2111–2124.

Cilibrasi, R. and Vitányi, P. M. B. (2005). Clustering by compression. *IEEE Transactions on Information Theory*, 5:1523–1545.

Crocomo, M. K. (2012). *Algoritmo de otimizaçao bayesiano com detecçao de comunidades*. PhD thesis, Universidade de São Paulo.

Dufourq, E. and Bassett, B. A. (2017). Text compression for sentiment analysis via evolutionary algorithms. In *Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), 2017*, pages 116–121. IEEE.

Felsenstein J. (2003). *Inferring phylogenies*. Sinauer Associates.

Giachanou, A. and Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2):28.

Kotzias, D., Denil, M., De Freitas, N., and Smyth, P. (2015). From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606. ACM.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6). arXiv: cond-mat/0309508.

Sanches, A., Cardoso, J. M. P., and Delbem, A. C. B. (2011). Identifying merge-beneficial software kernels for hardware implementation. In *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, pages 74–79, Cancun. IEEE.

Soares, A. H. M. and Delbem, A. C. B. Detecção de correlação em dados complexos usando ncd.

Valdivia, A. M. C. (2007). *Mapeamento de dados multidimensionais usando árvores filogenéticas: foco em mapeamento de textos*. PhD thesis, Universidade de São Paulo.

Ziegelmayer, D. and Schrader, R. (2012). Sentiment polarity classification using statistical data compression models. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 731–738. IEEE.