MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIANO - CAMPUS RIO VERDE

FABIANO MOREIRA DA SILVA

RECONHECIMENTO DE PADRÕES POR COMPRESSÃO APLICADO AO PROCESSAMENTO DA LINGUAGEM NATURAL

RIO VERDE
2019

FABIANO MOREIRA DA SILVA

RECONHECIMENTO DE PADRÕES POR COMPRESSÃO APLICADO AO PROCESSAMENTO DA LINGUAGEM NATURAL

Projeto do Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Rio Verde ligado ao Ministério da Educação, como requisito parcial da disciplina Tópicos de Pesquisa em Computação para a obtenção do título de Bacharelado em Ciência da Computação.

Orientador: Ms. Adriano Soares de Oliveira Bailão
Instituto Federal Goiano

RIO VERDE
2019

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

S586r

Silva, Fabiano Moreira da
Reconhecimento de Padrões por Compressão aplicado ao Processamento da Linguagem Natural / Fabiano Moreira da Silva;orientador Adriano Soares de Oliveira Bailão. -- Rio Verde, 2019.
44 p.

Monografia (em Ciência da Computação) -- Instituto Federal Goiano, Campus Rio Verde, 2019.

1. Processamento da Linguagem Natural. 2. Reconhecimento de Padrões. 3. Compressão. 4. Inteligência Artificial. I. Bailão, Adriano Soares de Oliveira, orient. II. Título.

Responsável: Johnathan Pereira Alves Diniz - Bibliotecário-Documentalista CRB-1 n°2376

INSTITUTO FEDERAL
Goiano

Repositório Institucional do IF Goiano - RIIF Goiano
Sistema Integrado de Bibliotecas

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnico-Científica

[ ] Tese
[ ] Dissertação
[ ] Monografia – Especialização
[x] TCC - Graduação
[ ] Produto Técnico e Educacional - Tipo: _____
[ ] Artigo Científico
[ ] Capítulo de Livro
[ ] Livro
[ ] Trabalho Apresentado em Evento

Nome Completo do Autor: Fabiano Moreira da Silva
Matrícula: 2016102192010455
Título do Trabalho: Reconhecimento de Padrões por Compressão aplicado ao Processamento da Linguagem Natural

Restrições de Acesso ao Documento

Documento confidencial: [x] Não [ ] Sim, justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: 10/02/2020
O documento está sujeito a registro de patente? [ ] Sim [x] Não
O documento pode vir a ser publicado como livro? [ ] Sim [x] Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

1. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
2. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
3. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Rio Verde, 10/02/2020.
Local Data

Fabiano Moreira da Silva
Assinatura do Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

Assinatura do(a) orientador(a)

INSTITUTO FEDERAL GOIANO – CÂMPUS RIO VERDE
DIRETORIA DE ENSINO
GERÊNCIA DE ENSINO DE GRADUAÇÃO

ATA DE DEFESA DO TRABALHO DE CURSO (TC)

ANO	SEMESTRE
2019	02

No dia 16 do mês de dezembro de 2019, às 10 horas e 45 minutos, reuniu-se a banca examinadora composta pelos docentes Adriano Soares O. Bailão, Douglas Cedrim Oliveira e Fábio Montanha Ramos, para examinar o Trabalho de Curso (TC) intitulado Reconhecimento de Padrões Por Compressão Aplicado ao Processamento da Linguagem Natural, do(a) acadêmico(a) Fabiano Moreira da Silva, Matrícula nº 2016102192010455 do curso de Ciência da Computação do IF Goiano – Câmpus Rio Verde. Após a apresentação oral do TC, houve arguição do candidato pelos membros da banca examinadora. Após tal etapa, a banca examinadora decidiu pela APROVADO do(a) acadêmico(a). Ao final da sessão pública de defesa foi lavrada a presente ata, que segue datada e assinada pelos examinadores.

Rio Verde, 16 de dezembro de 2019

Nome:
Orientador(a)

Nome:
Membro

Nome:
Membro

Observação:
( ) O(a) acadêmico(a) não compareceu à defesa do TC.

À minha família, por sua capacidade de acreditar e investir em mim. À minha mãe, pela dedicação e cuidado que foram dados, em alguns momentos, a esperança para seguir. À meu pai, pela presença que significou segurança e a certeza de que não estou sozinho nessa caminhada. Também aos meus irmãos e amigos, pelas alegrias, tristezas e dores compartilhadas.

AGRADECIMENTOS

Agradeço à Deus, por ter me fornecido energia e recursos para sempre seguir adiante nessa jornada. Agradeço aos meus pais. Minha mãe por sempre me apoiar em minhas tomadas de decisões, ao meu pai pelo exemplo de homem que é, por estar sempre comigo, independente da condição. Aos meus irmãos Vânderson e Uenes por estarem sempre próximos e dispostos a ajudar em qualquer dificuldade. Agradecimento especial ao meu melhor amigo Gustavo, por estar comigo ao longo de toda essa longa e árdua jornada.

Agradeço ao meu orientador Ms. Adriano Soares de Oliveira Bailão por aceitar me orientar de prontidão e contribuir de forma excepcional para com esta pesquisa.

Agradeço ainda a todo Instituto Federal Goiano, campus Rio Verde por fornecer e manter uma educação de qualidade.

Sou grato à Faculdade Almeida Rodrigues (FAR) que de forma indireta contribuiu para essa formação. Meus agradecimentos ainda para os colegas de curso e todos que direta ou indiretamente fizeram parte dessa etapa.

RESUMO

SILVA, Fabiano Moreira da. Reconhecimento de Padrões por Compressão aplicado ao Processamento da Linguagem Natural. 2019. 44 f. Projeto do Trabalho de Conclusão de Curso – Bacharelado em Ciência da Computação, Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Rio Verde. Rio Verde, 2019.

Esta pesquisa tem o intuito de contribuir para a comunidade de inteligência artificial com o processamento da linguagem natural, tema esse que vem ganhando cada vez mais força juntamente com a expansão da internet pelo mundo. Sua importância para a computação se dá através da necessidade de haver uma comunicação e compreensão entre o homem e a máquina. A pesquisa propõe uma extensão do framework DAMICORE aplicando algoritmos de compressão para auxiliar no reconhecimento de padrões e ser capaz de agrupar textos com alta porcentagem de acerto.
Palavras-chave: Processamento da linguagem natural, Reconhecimento de padrões, Compressão, Inteligência Artificial.

ABSTRACT

SILVA, Fabiano Moreira da. Pattern Recognition by Compression applied to Natural Language Processing. 2019. 44 f. Projeto do Trabalho de Conclusão de Curso – Bacharelado em Ciência da Computação, Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Rio Verde. Rio Verde, 2019.

This research aims to contribute to the artificial intelligence community with a natural language processing, which is gaining strength with the expansion of the Internet around the world. Its importance for computing is through the need for communication and understanding between man and machine. The research proposes an extension of the DAMICORE framework applying compression algorithms to help in the recognition of patterns and to be able to group texts with high percentage of correctness.

Keywords:Natural Language Processing, Pattern Recognition, Compression, Artificial Intelligence

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RP	Reconhecimento de Padrões
DAMICORE	Data-Mining of Code Repositories
PLN	Processamento da Linguagem Natural
TT	Teste de Turing
LN	Linguagem Natural
SRI	Sistemas de Recuperação da Informação
TA	Tradução Automática
LZW	Lempel-Ziv-Welch

SUMÁRIO

1 INTRODUÇÃO

Atualmente, existem diversas formas de reconhecimento de emoções do usuário por meio de padrões, podendo ser citadas como as principais o reconhecimento de expressões faciais, comportamentos observáveis, entonação vocal e sinais fisiológicos. Desse modo, o presente estudo visa observar o reconhecimento de emoções por padrões textuais, o qual é recorrente e ganha muita força com no cenário atual da internet.

No processo de análise de dados os profissionais desse segmento devem ser capacitados, tanto do conhecimento estatístico, quanto do domínio do problema a ser tratado. O auxílio de ferramentas necessita ter cada vez mais o domínios específicos, possibilitando menor interferência do analista no processo de análise a partir desta fase de tratamento genérico dos dados.

O framework DAMICORE é uma ferramenta que utiliza aproximações chamadas de *compactadores* para a medição de informações em objetos de fontes de dados não estruturados. *Compactadores* são técnicas adaptativas e estatísticas de compressão *sem perdas*. Embora, os frameworks baseados em *compactadores* possuam alta capacidade de generalização para o tratamento das fontes de dados em aplicações de Reconhecimento de Padrões (RP), o modelo de representação dos objetos de dados com base nesse tipo de aproximação torna a abordagem dependente do algoritmo que o compactador encapsula, levando a implicações como por exemplo, ambiguidade na representação das amostras, falta de precisão na classificação e resultados indeterminísticos.

A compressão de dados é uma estratégia que pode ser usada para auxiliar na classificação, resolvendo o problema de um texto com muitas características repetidas e consequentemente, auxiliar na diminuição do custo computacional, dessa forma colabora,pois é um fator para a redução de dimensionalidade.

Já os algoritmos de compressão como Huffman e LZW contam com a similaridade de características no conjunto fornecido para ter uma boa compactação, portanto, quando se comprimem, simultaneamente é encontrado padrões vistos antes no conjunto de caracteres. Por meio dessa aplicação pretende-se verificar resultados de classificação.

Esta pesquisa propõe a extensão do fluxo de operação do framework DAMICORE (*Data-Mining of Code Repositories*) de forma a direcionar o *bias* de uma aplicação de RP (Reconhecimento de Padrões) a partir de uma fase chamada *Granulação* e outra fase chamada *Codificação* além da extensão do modelo para a aplicação de técnicas de compressão *sem perdas*.

Ainda é proposto o uso de técnicas de pré-processamento textual, como remoção de *stopwords*, radicalização de palavras e contagem da frequência para analisar a relevância do vocábulo no contexto, com o intuito de minerar ainda mais os dados a serem processados, de tal modo, trabalhar diferentes abordagens do algoritmo LZW a fim de adaptar melhor

ao contexto aplicado e obter excelentes resultados.

Entretanto,o resultado em processos de RP como formação de agrupamentos e classificação, podem incorporar em seus modelos a análise de fontes de dados estruturados e semi estruturados, ganho qualitativo na formação e visualização dos agrupamentos e melhora da delimitação das fronteiras (limites de decisão) entre grupos de objetos.

De tal modo os objetivos que se almejam alcançar com esta pesquisa são interpretar mensagens de texto a partir da compreenção dos aspectos léxicos, sintáticos e semânticos em gêneros textuais.

Os objetivos específicos são definir o modelo de representação do conjunto de mensagens na forma de objetos de dados, produzir o modelo de aprendizado não-supervisionado para a definição do contexto dos objetos de dados e criar o modelo de decisão baseado em aprendizado supervisionado para a classificação de objetos mensagens.

O problema computacional relacionado a situação que envolve esta pesquisa é o de classificação de texto, que soluciona casos como identificação de idiomas, análise de sentimentos, classificação do gênero, entre outras situações. Geralmente é usado como estratégia um algoritmo de aprendizado supervisionado contando com as características, aqui sendo as palavras, e o valores (quantidade de vezes que a palavra se repete), nessa estratégia o custo computacional pode ser demasiadamente alto de acordo com o número de características e o uso de um algoritmo supervisionado o qual exige que a classificação seja feita somente após um treinamento prévio.

A linguagem natural é passível de ambiguidade, dado o fato de que uma mesma palavra pode emitir dois ou mais sentidos diferentes de acordo com o contexto em que se encontra na sentença. Possui um vocabulário bastante extenso, diferentemente das máquinas e outros seres vivos. De tal modo, torna-se um trabalho computacional de grande relevância, ao compreender que as máquinas possam processar a linguagem natural estabelecendo uma relação de comunicação entre as máquinas e os humanos.

Este trabalho está organizado da seguinte maneira: o Capítulo 2 traz conceitos sobre Processamento da Linguagem Natural, aprendizagem de máquina, algoritmos de compressão e reconhecimento de padrões. O capítulo 3 traz os materiais e a metodologia usada. Seguindo o Capítulo 4 traz os resultados obtidos e análise feita sobre a pesquisa. Enfim, no Capítulo 5 é apresentado as considerações finais e trabalhos futuros. Após o Capítulo 5 há a lista de referências úteis, que auxiliaram no embasamento teórico para a realização desta.

2 REVISÃO DE LITERATURA

Conceitos como processamento da linguagem natural, pré-processamento de textos, classificação automática, aprendizagem de máquina, algoritmos de compressão (Huffman e LZW) e reconhecimento de padrões serão melhor definidos nos próximos tópicos deste capítulo.

2.1 Processamento da Linguagem Natural (PLN)

Segundo Russell e Norvig (2013), há mais de um trilhão de páginas de informações na Web, quase todas elas em linguagem natural. Um agente que deseja adquirir conhecimento precisa entender (pelo menos parcialmente) a ambígua confusa linguagem que os seres humanos usam.

"O processamento da linguagem natural pode ser vista como a forma de comunicação entre o homem e a máquina, sendo essa comunicação em qualquer linguagem que se fale" (TURBAN et al., 2010 apud BRITO, 2016).

Portanto, para uma máquina se comunicar com um ser humano é necessário haver uma tradução entre as linguagens interpretadas por cada um, nesse contexto entra o processamento da linguagem natural, onde técnicas e algoritmos da Ciência da Computação são aplicadas à gramática humana a fim de ser capaz de compreender a semântica da linguagem, possibilitando receber dados de entrada na linguagem humana, processá-los e gerar dados de saída compreensíveis.

2.1.1 Surgimento da PLN

Com base em Perna (2010) a área de PLN começou com tentativas de tradução automática na segunda metade da década de 1940. Esses trabalhos iniciais estavam relacionados com esforços prévios de quebra de códigos durante a Segunda Guerra Mundial. De um ponto de vista teórico, esses trabalhos inicias em tradução automática estavam baseados na criptografia e teoria da informação.

Segundo Setzer (2006) Allan Turing teve uma contribuição prática para o nascimento da PLN em meados de 1950, a partir da aplicação do seu chamado Teste de Turing (TT), que ao invés de responder à pergunta "pode-se ter computadores inteligentes?" Ele formulou seu teste, que se tornou praticamente o ponto de partida da pesquisa em "Inteligência Artificial". O teste consiste em se fazer perguntas a uma pessoa e um computador escondido, sem que ambos saibam. O TT propõe que uma máquina compreenda e simule a linguagem natural humana.

2.1.2 Definição

Desse modo Reese e Bhatia (2015) trazem a definição formal de PNL frequentemente inclui texto no sentido de que é um campo de estudo usando Ciência da Computação, inteligência artificial e linguística formal conceitos para analisar a linguagem natural. Outra definição, sugere que é um conjunto de ferramentas usadas para obter informações úteis e significativas da linguagem natural fontes como páginas da web e documentos de texto.

Para Perna (2010) o Processamento de Linguagem Natural (PLN) é um campo estudado pela Ciência da Computação que proporciona o desenvolvimento de softwares capazes de analisar, reconhecer e gerar textos em linguagens humanas, ou linguagens naturais.

2.1.3 Etapas

Ao decorrer da pesquisa serão aplicadas diversas técnicas utilizadas para tentar derivar o significado de um documento textual. Nesta seção serão conceituadas técnicas utilizadas no meio da PLN para abstrair ao máximo o conteúdo semântico de um texto de forma que os computadores possam compreender e interpretar os dados gerados. Será visto como o texto pode ser dividido em sua menor unidade, processo chamado de granulação, assim como esses elementos podem ser categorizados.

2.1.3.1 Análise Léxica

Resume-se que a principal função da análise léxica é a *tokenização*. *Tokens* são definidos como os elementos significativos que são gerados, usando técnicas de análise lexical. Nesta pesquisa temos como exemplo de *tokenização* a seguinte situação: tendo como entrada o texto "*I have to jiggle the plug to get it to line up right to get decent volume*" é retornado um vetor com os *tokens*: [*"I", "have", "to", "jiggle", "the", "plug", "to", "get", "it", "to", "line", "up", "right", "to", "get", "decent", "volume"*]. Após os *tokens* será gerado um sistema de PLN dos quais a partir desse momento podem avançar à próxima etapa.

> A análise lexical é definida como o processo de decompor um texto em palavras, frases e outros elementos significativos. Dessa forma, a análise lexical é baseada na análise no nível da palavra. Nesse tipo de análise, também é observado o significado das palavras, frases e outros elementos, como símbolos(THANAKI, 2017).

2.1.3.2 Análise Morfológica

Na etapa morfológica acontece o processamento que produz a separação das palavras em função de sufixo e prefixo. Portanto, ao final do processo é armazenado somente o radical da palavra em questão.

"Dessa maneira, a morfologia se preocupa com a estrutura interna das palavras. Em particular, inclui o estudo da estrutura de palavras e como essa estrutura muda em diferentes contextos" (GONÇALVES et al., 2006).

Rich et al. (1994) afirma que esta etapa consiste em analisar a forma da palavra mostrando assim os vários elementos que lhe constituem. Exemplos de elementos mórficos são: radical, raiz e terminações. Ao se isolar uma palavra a mesma deve ser analisada pelo seu componente. Assim, ao se informar o texto "Abrir o arquivo teste.txt", a análise morfológica tem que saber que teste.txt é um arquivo, e que ".txt" é a extensão desse arquivo.

Ao final do processo pode-se afirmar que a análise morfológica é a responsável por categorizar de forma isolada cada palavra de acordo com que rege as regras de classificação de cada idioma. Como exemplo temos a língua portuguesa que pode classificar as palavras em: substantivo, adjetivo, numeral, verbo, advérbio, pronome, artigo, preposição, conjunção e interjeição.

2.1.3.3 Análise Sintática

Destarte ao trabalhar com sentenças textuais é comum o uso da análise sintática, que em um determinado idioma refere-se às regras que validam a estrutura de uma frase.

Na subseção anterior foi visto uma análise a nível de palavra. Nesta subseção o texto será analisado em um nível superior, na sintaxe o texto é compreendido a partir da gramática e estrutura das frases.

Thanaki (2017) define a análise sintática como a análise que nos diz o significado lógico de determinadas sentenças ou partes dessas. É preciso salientar que é necessário regras gramaticais para definir o significado lógico e obter a correção das sentenças.

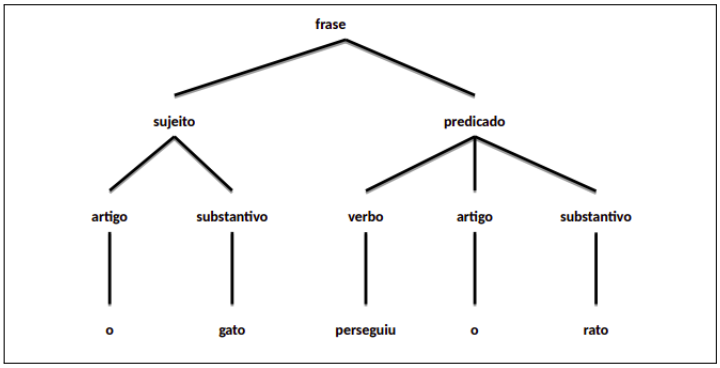
Thanaki (2017) ainda diz que a análise sintática é uma área bem desenvolvida da PLN a qual lida com a sintaxe da linguagem natural (LN), assim na análise sintática, regras gramaticais são usadas para determinar quais sentenças são legítimas.

A Figura 1 ilustra uma estrutura para análise a partir de uma árvore sintática.

2.1.3.4 Análise Semântica

Oliveira (2011) classifica a semântica em duas partes, sendo elas: léxica e gramatical, a semântica léxica busca descrever o sentido através do uso da decomposição semântica das unidades léxicas ou através das redes semânticas que leva em consideração, o fato de como os humanos memorizam e categorizam os conceitos. Já a semântica gramatical tenta buscar o sentido através de uma fórmula lógico-semântica, porém há casos que em uma estrutura pode dar origem a duas representações semânticas, como em "uma professora de capoeira pernambucana" pode referir-se a uma pessoa nascida em Pernambuco ou a uma pessoa que ensina capoeira no estilo pernambucano.

Figura 1 – Árvore sintática



Fonte: Silva (2013)

> A semântica de uma frase é o seu significado. Como falantes da língua portuguesa, entende-se o significado da frase "João bateu na bola", já ao ser lido ou pronunciado. No entanto, português e outros idiomas podem ser ambíguos às vezes e o significado de uma frase pode ser apenas determinado a partir do seu contexto(BORDIGNON et al., 2018).

2.1.3.5 Análise Pragmática

O processo de análise pragmática consiste em reinterpretar frase a fim de buscar o seu real significado. Um exemplo que necessita desse tipo de análise é quando há uma pergunta: "Você sabe com quem está falando?" Nesse caso, é função do analisador, definir se o locutor deseja saber se o ouvinte sabe com quem está falando, ou se é uma fala intimidadora do locutor.

A análise é exigida somente quando há casos com ambiguidade no sentido. Ainda é importante atentar-se que a análise pragmática não se restringe somente a uma frase, ela procura o significado dentro de todo o contexto em que está inserida.

2.1.3.6 Integração do Discurso

"A integração do discurso está intimamente relacionada à pragmática. Considera-se o contexto maior para qualquer parte menor da estrutura da LN. Desse modo, LN é tão complexa e, na maioria das vezes, sequências de textos dependem do discurso anterior" (THANAKI, 2017).

> Nessa perspectiva, esse conceito ocorre frequentemente em ambiguidade pragmática. Portanto, a análise trata de como a sentença imediatamente anterior, pode afetar o significado e a interpretação da sentença seguinte. De modo que texto pode ser analisado em um contexto maior, como nível de parágrafo, nível de documento e assim por diante (THANAKI, 2017).

2.1.4 Utilidades

A PLN é significativa e útil, isso implica que ela tem valor comercial, embora comumente seja utilizada para fins de pesquisa científica no meio acadêmico. Seu valor pode ser facilmente visto no apoio à motores de busca. Uma consulta de usuário é processada usando técnicas de PLN para gerar uma página de resultado que um usuário possa usar. Os mecanismos de pesquisa modernos têm sido muito bem-sucedidos a respeito disso. As técnicas de PLN também encontraram uso em sistemas automatizados de ajuda e no suporte a sistemas de consulta complexos.

A seguir será descrito diversos tipos de sistemas que empregam o uso da PLN.

2.1.4.1 Sistemas de Recuperação da Informação (SRI)

Conforme Monteiro et al. (2017 apud ARAUJO, 1995) os Sistemas de Recuperação da Informação devem representar, armazenar, organizar e localizar os itens de informação, o referido autor aponta que a indexação é a principal função de um SRI, e seus componentes devem incluir: documentos; necessidades do usuário; consulta formulada e o processo de recuperação propriamente dito.

2.1.4.2 Chatbot

Em suma *chatbots* são como agentes que simulam o comportamento humano em parâmetros de linguagem natural e logo auxiliam com informações ou ações.

> Os *Chatbots* são normalmente desenvolvidos para tratar sobre uma área de conhecimento específico através de sentenças da linguagem que se deseja abordar. Dessa maneira, são acionados através de uma pergunta direta do usuário ou, quando programado para isso, procuram comentários sobre o assunto e os responde (HUANG; ZHOU; YANG, 2007).

2.1.4.3 Tradução Automática

Caseli (2017) define que a Tradução Automática (TA) é tanto uma das principais subáreas do Processamento Automático de Línguas Naturais (PLN) como uma aplicação computacional disponível em sites e sistemas/aplicativos para computadores e dispositivos móveis (celulares, tablets, etc.). Na TA, a partir da informação fornecida como entrada em um idioma original (língua fonte), um site/sistema/aplicativo gera uma versão equivalente em outro idioma (língua alvo).

2.1.4.4 Sumarização Automática de Textos

A sumarização é considerada uma técnica que lida com resumos textuais de forma automática, onde um computador fica responsável por produzir uma versão resumida do texto original.

Souza et al. (2017) explica que é comum a ausência de um resumo e de palavras-chave como elementos pré-textuais a auxiliarem o leitor em seu primeiro contato com um texto na Web, ambiente onde é comum haver uma grande quantidade de textos com conteúdos semelhantes. Nesse caso, uma forma de contribuir para a melhoria da representação, mediação, recuperação e uso da informação seria prover as páginas Web de um importante metadado: o resumo, o que permitiria um primeiro contato do leitor com as páginas Web muito mais produtivo, pois a partir dele é possível decidir ler o documento na íntegra ou não. Nesse ínterim, o resumo é efetivamente um elemento de acessibilidade: um caso particular de acessibilidade informacional.

2.1.5 Pré-processamento Textual

O pré-processamento textual é a primeira etapa após a coleta dos dados, os quais quando coletados geralmente não estão prontos para serem processados, necessitando de um tratamento anterior, com o intuito de auxiliar no custo computacional do processamento propriamente dito, assim como na acurácia dos resultados.

"Observa-se que o principal objetivo de pré-processar um texto, consiste na filtragem e limpeza dos dados, eliminando redundâncias e informações desnecessárias para o conhecimento que se deseja extrair" (GONÇALVES et al., 2006).

Por conseguinte, na fase de pré-processamento há etapas de extrema importância para redução do custo computacional e para boa acurácia nos resultados, etapas como *tokenização*, remoção de *stopwords* e redução do léxico.

A *tokenização* trata-se dos grânulos elementares nos textos em seu conceito formal, são as palavras. Com a implementação do *framework* DAMICORE os tokens poderão sofrer pequenas alterações, onde os grânulos elementares poderão ser representados pela unidade mínima da gramática, os caracteres.

"Por fim, o principal objetivo de criar *tokens* é a tradução de um texto em dimensões possíveis de se avaliar, analisar, para obtenção de um conjunto de dados estruturados" (JACKSON; MOULINIER, 2007).

A remoção de *stopwords* é essencial para o custo computacional, dado fato que esta etapa trata de remover palavras as quais não agregam significado aos textos, são palavras vazias que dão concordância e ligação às frases. É essencial a atenção na lista de palavras que compõem a *stopword*, pois isso acarretará na perda de informação do texto original.

Na análise de dados a **maldição da dimensionalidade** é um grande problema tanto no impacto da desempenho quanto no resultado final. A redução do léxico ganha aqui sua relevância, pois propõe reduzir a quantidade de características a serem processadas usando a normalização de palavras, processo este que consiste em extrair das palavras "tokenizadas" seus radicais, como exemplo as palavras *casa*, *casinha*, *casebre*, *casarão* que derivam todas elas do radical *cas*. Ainda na redução do léxico utiliza-se remoção de

stopwords que se trata de remover palavras vazias, que é uma palavra não que tem grande relevância semântica no contexto, como artigos, preposições, etc.

Além da remoção de *stopwords* foram aplicadas as seguintes técnicas de pré-processamento textual:

- Tag de negação: esta técnica é utilizada quando se encontra palavra com sentido de negação, a partir de então insere-se uma palavra de negação a frente de todas as outras até que se termine a sentença com algum ponto. Está técnica é aplicada após a remoção a remoção de *stopwords*. Por exemplo, temos como entrada o texto "*Eu não estou rico ainda. Irei trabalhar até ficar rico*" gera como saída o seguinte texto: "*'não-rico', 'trabalhar', 'ficar', 'rico'*".
- Combinação com bigramas: Nesta técnica as palavras serão combinadas 2 (duas) a 2 (duas). Essa técnica também é aplicada após a remoção de *stopwords*. Por exemplo, temos como entrada o texto "*Eu não estou rico ainda. Irei trabalhar até ficar rico*" gera como saída o texto com as seguintes combinações: "*('rico', 'trabalhar'), ('trabalhar', 'ficar'), ('ficar', 'rico')*" .
- Frequência de palavras: Nesta técnica as palavras passam por um contagem de forma a definir com que frequência cada uma delas ocorre. A partir disso é tirado uma média geral com base em $\frac{qtde\ total\ palavras}{qtde\ palavras\ distintas}$ e a palavra só retornará como texto de saída caso a sua ocorrência seja maior ou igual à média.

2.2 Aprendizagem de Máquina

O aprendizado de máquina trata-se da capacidade das máquinas "aprenderem" a partir de algoritmos que tem o intuito de deduzir e extrair padrões em grande quantidade de dados.

"Da mesma forma, o aprendizado de máquina visa estudar como os algoritmos computacionais são capazes de automaticamente melhorarem a execução de tarefas através da experiência. Os algoritmos desenvolvidos sobre a aprendizagem de máquina se baseiam na estatística e probabilidade para aprender padrões complexos a partir de algum corpus" (MITCHELL, 1997 apud BRITO, 2016).

Nesse âmbito, entende-se que a aprendizagem de máquina geralmente se estende em três vertentes, sendo elas: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço. Nesta pesquisa será aplicada a técnica de aprendizagem de máquina não supervisionada.

Segundo Russell e Norvig (2013) na aprendizagem não supervisionada, o agente aprende padrões na entrada, embora não seja fornecido nenhum *feedback* explícito. A tarefa mais comum de aprendizagem não supervisionada é o agrupamento: a detecção de grupos de exemplos de entrada potencialmente úteis. Por exemplo, um agente de táxi pode desenvolver gradualmente um conceito de "dia de tráfego bom" e "dia de tráfego ruim" sem nunca ter sido rotulados anteriormente, tornando modelo de cada um deles pelo professor.

Russell e Norvig (2013) ainda descrevem que na aprendizagem supervisionada, o agente observa alguns exemplos de pares de entrada e saída, e aprende uma função que faz o mapeamento da entrada para a saída.

Walaa, Ahmed e Hoda (2014) salientam que para resolver o problema de classificação de textos, são comumente utilizados algoritmos baseados em aprendizado de máquina fazendo uso de traços linguísticos. Dentro deste aspecto, também é utilizado o aprendizado supervisionado, o qual consiste em um método que depende de documentos preestabelecidos, contendo regras de linguagem para realizar a classificação.

2.3 Algoritmos de Compressão

Desse modo, os algoritmos de compressão podem se dividir em dois tipos, compressão com perda e sem perda. Nesta pesquisa os que nos interessam são os algoritmos de compressão sem perda, dado o fato de que a perda de caracteres pode afetar a significado dos textos.

> Compressão de dados é uma forma de usar menos espaço para armazenar a maior quantidade de informação, podendo assim reduzir o espaço utilizado, sendo este uma imagem, um texto, um vídeo ou qualquer tipo de arquivo. Para diminuir o espaço utilizado, usa-se diversos algoritmos cujo objetivo é reduzir a quantidade necessária de *Bytes* para representar uma determinada informação (UFSM, 2006).

Os algoritmos abordados aqui são Huffman e LZW (Lempel-Ziv-Welch).

2.3.1 Algoritmo de Compressão Huffman

O código de Huffman foi desenvolvido por David Albert Huffman em meados de 1952 em sua pesquisa de doutorado com o propósito de criar um método capaz de comprimir dados sem que se perca informação no ato de descompactação.

> Considera-se a existência de um alfabeto fonte, cujos símbolos têm pesos determinados. O objetivo será a construção de um código de prefixo minimal que seja de redundância mínima. Sendo assim, tem por princípio associar códigos menores a símbolos mais prováveis e códigos maiores a símbolos menos prováveis. O código necessariamente terá comprimento variável, enquanto que os símbolos podem apresentar comprimento fixo ou variável (SOUZA, 1991).

O código de Huffman é baseado em redundância mínima, onde as características que se repetem com maior frequência são representados em uma quantidade menor de *bits*.

> Para compressão de texto pode ser utilizado o conceito de substituição de palavras por *bits*. Isso pode resultar em uma compressão maior na maioria dos casos. Se analisarmos o caractere “a” que em um texto pode representar uma palavra, o caractere por si só tem o tamanho de oito *bits* (SANTOS, 2018).

Com intuito de exemplificar temos a palavra *banana* que sem usar compactação possui 6 Bytes (48 bits), devido ao fato de cada carater ocupar 1 Byte, após aplicar a codificação de Huffaman o tamanho de cada carater diminui conforme ilustra a Tabela 1, passando a ocupar agora 13 bits.

Tabela 1 – Exemplo de codificação de Huffman

Caracter	**Frequência**	**Codificação**
b	1	101
a	3	01
n	2	00

Fonte: O próprio autor

2.3.2 Algoritmo de Compressão LZW

O algoritmo de codificação baseado em dicionários baseado em combinações (padrões) tem suas origens a partir de 1977 com o algoritmo LZ77 e ganha uma evolução em 1978 com e assim foi renomeado para LZ78. Em 1989 Terry Welch propõe um aperfeiçoamento no processo de codificação.

> Em 1977, Abraham Lempel e Jakob Ziv publicaram um artigo com um algoritmo universal para compressão de dados. O algoritmo foi chamado inicialmente de LZ77. Em 1978, Lempel e Ziv introduziram um melhoramento, um esquema de compressão baseado em dicionário, chamado de LZ78 (CUNHA, 20-).
>
> Em 1989, Terry Welch propôs um novo esquema de compressão substitucional chamado Lempel-Ziv-Welch (LZW), o qual é principalmente baseado no LZ78. Esta mudança aperfeiçoa o processo de codificação excluindo o caractere ao fim de cada código quando possível(CUNHA, 20-).

Segundo Brazil (20-) o algoritmo LZW é um algoritmo de codificação de palavras, cuja ideia principal é ir construindo um dicionário de símbolos ou palavras conforme o texto ou a informação vai sendo processado pelo algoritmo. No início da codificação de um texto (ou imagem), o dicionário LZW possui inicialmente como palavras apenas os 256 caracteres da tabela ASCII (ou, no caso de uma imagem, a quantidade de tons que compõe a paleta da imagem).

Brazil (20-) sob o mesmo ponto de vista, afirma que no processo de análise do texto, o dicionário vai criando novas palavras a partir de combinações de pelo menos 2 caracteres encontrados no texto. A cada nova combinação encontrada, é criada uma nova entrada no dicionário que será utilizada para codificações futuras. Ao final do processo de codificação, teremos um dicionário de combinações de palavras que poderá ser utilizado para codificar o texto, possivelmente reduzindo a quantidade de espaço gasto para seu armazenamento.

Considerando a codificação do termo *BABABABABABAB* o dicionário resultante é apresentado na Tabela 2, de modo que o termo codificado passa a ser os índices 1,0,3,5,4,7,1.

Tabela 2 – Exemplo de codificação LZW

Índice	**Sequência de valores**
0	A
1	B
2	C
3	BA
4	AB
5	BAB
6	BABA
7	ABA
8	ABAB

Fonte: O próprio autor

Portanto, fica assegurado que o algoritmo LZW trabalha com um dicionário capaz de representar padrões já encontrados no texto.

2.4 Reconhecimento de Padrões

> A área de reconhecimento de padrões é importante devido às ocorrências na vida humana tomarem formas de padrões. A formação da linguagem, o modo de falar, o desenho das figuras, o entendimento das imagens, tudo envolve padrões diferenciados. Por outro lado, o RP é uma tarefa complexa, a qual busca sempre, avaliar as situações em termos dos padrões das circunstâncias que as constituem, descobrir relações existentes no meio, para melhor entendê-las e adaptar-se (PAO, 1989 apud MORAIS, 2010).

"O reconhecimento de padrões está relacionado aos aspectos associados a problemas específicos, como por exemplo, redução de dimensionalidade, extração de características e escolha de um classificador adequado" (CAMPOS, 2001 apud MORAIS, 2010).

Para Monteiro et al. (2008) um sistema para reconhecimento de padrões engloba três grandes etapas: representação dos dados de entrada e sua mensuração, extração das características e finalmente identificação e classificação do objeto em estudo.

Monteiro et al. (2008) descreve as etapas para um sistema de reconhecimento de padrões da seguinte maneira: a primeira refere-se à representação dos dados de entrada que podem ser mensurados a partir do objeto a ser estudado. A segunda etapa consiste na extração de características intrínsecas e atributos do objeto e consequente redução da dimensionalidade do vetor padrão. É a fase da extração das características. A terceira etapa em reconhecimento de padrões envolve a determinação de procedimentos que possibilitem a identificação e classificação do objeto em uma classe de objetos.

Monteiro et al. (2008) ainda relata que quando os padrões de uma classe são vetores cujos componentes são números reais, a classe do padrão pode ser estabelecida segundo formas do agrupamento, *clusters*, de tais pontos no plano. Havendo uma separação entre os pontos de forma clara, técnicas como, distância-mínima podem ser utilizadas. Caso haja superposição entre os *clusters*, técnicas mais elaboradas são necessárias, tais como, classificação por funções similares (métodos estatísticos), por treinamento de padrão (métodos determinísticos) ou outros algoritmos mais adequados.

3 MATERIAIS E MÉTODOS

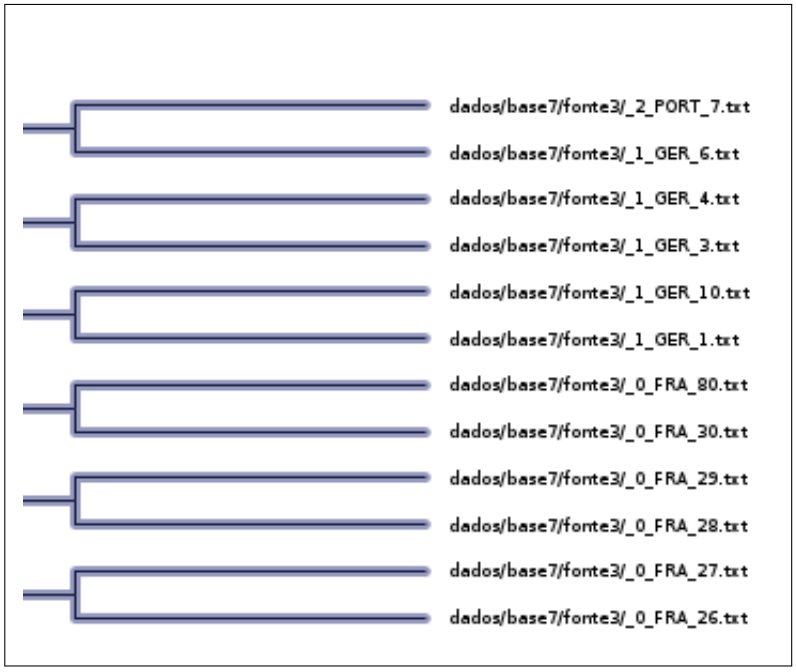
Neste capítulo será apresentado todos os materiais, ferramentas e métodos necessários para realizar esta pesquisa.

3.1 Coeficiente

Para analisar os resultados desta pesquisa foi desenvolvido um coeficiente denominado de: *Coeficiente G*, capaz de medir a taxa de acerto com base na árvore gerada.

Considerando a árvore gerada na Figura 2, é montada uma outra árvore ligando os nós folhas e caso a aresta compartilha de nós com a mesma classe esta aresta recebe peso 1, caso contrário recebe peso 0. Após esse processo a árvore da Figura 2 é transformada na árvore da Figura 3.

Figura 2 – Árvore gerada como resultado



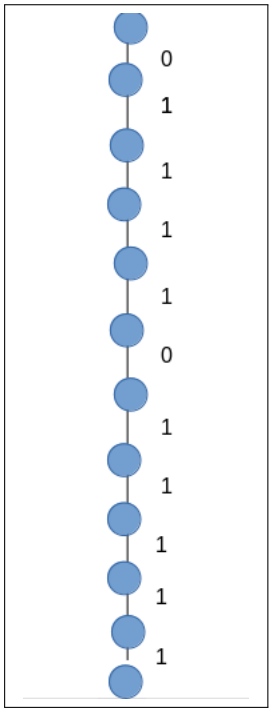
Fonte: O próprio autor

Para o *Coeficiente G* é considerada a fórmula Equação (1):

$$G = \frac{\sum_{i=1}^{A} a_i}{(V-1)-(C-1)} \tag{1}$$

Considere A como o total de arestas, V como total de vértices, C como total de classes e a_i como peso de uma determinada aresta. Desta forma fica V fica restrito a ser maior do que C, caso V seja igual a C a árvore é dada como já agrupada.

Figura 3 – Árvore para calcular coeficiente G



Fonte: O próprio autor

Após simplificar o denominador o resultado obtido é mostrado na fórmula Equação (2)

$$G = \frac{\sum_{i=1}^{A} a_i}{V - C} \tag{2}$$

3.2 Dados

Nesta pesquisa utiliza-se fonte de dados no formato de texto disponíveis na *web* retirados dos seguintes repositórios: *UCI Machine Learning Repository*, *Twitter*, *arxiv.org*, IMDB com sinopses de filmes e outros diversos sites nacionais contendo notícias de diversas categorias.

No total foram coletadas 1.162 textos, sendo eles pertencentes as seguintes categorias:

- Sinopses de Filmes IMDB (45)
- Resumos de artigos do Arxiv.org (250)
- Comentários do e-commerce Amazon (200)
- Emoções no Twitter (300)
- Notícias do Brasil (Twitter) (300)
- Artigos Noticiários (107)
- Detecção de Idiomas (260)

Os dados são explanados com mais detalhes nas subseções seguintes.

3.2.1 Sumarização

Nesta seção é descrita a coleta de dados onde os textos se encaixam na categoria de resumos de um arquivo textual como um todo.

3.2.1.1 Sinopses de Filmes IMDB

A princípio coletou-se um total de 45 resumos de filmes da base de dados do IMDB, divididos em 7 classes, conforme mostrado na Tabela 3.

Tabela 3 – Dados coletados do IMDB

Classe	**Objetos**
AÇÃO	9
BIOGRAFIA	7
COMÉDIA	6
DRAMA	7
HISTÓRIA	4
ROMANCE	6
GUERRA	6
Total	**45**

Fonte: O próprio autor

Como amostra de um objeto da classe ação temos a seguinte sinopse: " *Indian army special forces carries a covert operation, avenging the killing of fellow army men at their base by a terrorist group.*". Todos os objetos de dados desta base estão no idioma inglês.

3.2.1.2 Resumos de artigos do Arxiv.org

Nesta etapa foram retirados do repositório Arxiv.org um conjunto com 250 amostras de *abstracts* de artigos separados em 5 classes, das quais encontram-se descritos na Tabela 4.

Tabela 4 – Dados coletados do Arxiv.org

Classe	**Objetos**
BIOLOGIA	50
QUÍMICA	50
CIÊNCIA DA COMPUTAÇÃO	50
FÍSICA	50
CIÊNCIAS SOCIAIS	50
Total	**250**

Fonte: O próprio autor

Temos como amostra de um objeto de dados o seguinte abstract da classe Biologia: "*Course-based undergraduate research experiences (CUREs) are a type of laboratory learning environment associated with a science course, in which undergraduates participate in novel research. According to Auchincloss et al. (CBE Life Sci Educ 2104; 13:29–40), CUREs are distinct from other laboratory learning environments because they possess five core design components, and while national calls to improve STEM education have led to an increase in CURE programs nationally, less work has specifically focused on which core components are critical to achieving desired student outcomes. Here we use a backward elimination experimental design to test the importance of two CURE components for a population of non-biology majors: the experience of discovery and the production of data broadly relevant to the scientific or local community. We found nonsignificant impacts of either laboratory component on students' academic performance, science self-efficacy, sense of project ownership, and perceived value of the laboratory experience. Our results challenge the assumption that all core components of CUREs are essential to achieve positive student outcomes when applied at scale.*". Todos os objetos desta base encontram-se no idioma inglês.

3.2.2 Análise de Sentimentos

Nesta seção contém, nas Tabelas 5, 6 e 7 são relacionadas as quantidades de objetos coletados com suas respectivas classes.

3.2.2.1 Comentários do e-commerce Amazon

Anteriormente realizou-se coletas dos comentários de compradores, dos quais demostraram seu nível de satisfação após adquirir produtos da loja virtual Amazon, dados esses recolhidos através do repositório *UCI* contendo um total de 200 comentários separados em 2 classes conforme detalha a Tabela 5.

Tabela 5 – Dados coletados da Amazon

Classe	**Objetos**
NEGATIVO	100
POSITIVO	100
Total	**200**

Fonte: O próprio autor

Todos os objetos contidos nesta base estão no idioma inglês. Exemplo de um objeto da classe positivo: "*This phone is slim and light and the display is beautiful.*"

3.2.2.2 Emoções no Twitter

Assim houve a coleta de 300 *tweets*, a partir do repositório *UCI*, formando uma base de dados composta por 6 classes distintas capazes de expressar emoções no conteúdo das postagens. Informações dos textos coletados é descrita na Tabela 6.

Tabela 6 – Dados coletados do Twitter

Classe	**Objetos**
TRISTEZA	50
SURPRESA	50
ALEGRIA	50
DESGOSTO	50
RAIVA	50
MEDO	50
Total	**300**

Fonte: O próprio autor

Como exemplo de um objeto desta base temos o seguinte *tweet* da classe tristeza: "*Is wanting @AxelLoitz to be healthy to join me back in the paint*" Todos os objetos desta base encontram-se no idioma inglês.

3.2.2.3 Notícias do Brasil (Twitter)

Logo após coletadas as notícias do cenário político e econômico brasileiro expressando em relação ao sentimento de positividade, negatividade e neutro, sendo um total de 300 textos. Os dados aqui obtidos foram retirados do repositório <https://github.com/minerandodados/mdrepo/blob/master/titulo_noticias.txt>. A Tabela 7 descreve como foi dividida a coleta.

Tabela 7 – Dados coletados do Twitter

Classe	**Objetos**
POSITIVO	100
NEGATIVO	100
NEUTRO	100
Total	**300**

Fonte: O próprio autor

Temos como exemplo um instância da classe positivo: "quina são joão paga quase sete apostas", sendo todas as instâncias desta base no idioma português.

3.2.2.4 Artigos Noticiários

Foram coletados um total de 107 artigos de notícias em diversos portais brasileiros categorizados cada um com sua especialidade. Os dados desta base foram retirados dos

seguintes portais: G1, Globo Esporte, R7, IG, Correio Brasiliense, El País e Tecmundo. No total foram 5 classes, mais detalhes são mostrados na Tabela 8.

Tabela 8 – Dados coletados de sites de notícias nacionais

Classe	**Objetos**
ECONOMIA	15
ESPORTE	21
POLÍTICA	17
SAÚDE	27
TECNOLOGIA	27
Total	**107**

Fonte: O próprio autor

A fim de exemplificar um objeto desta base temos a seguinte notícia da classe economia:

Caminhoneiros vão parar em 21 de maio se diesel subir, diz líder Por Leila Souza Lima | Valor SÃO PAULO - Há ampla mobilização para uma paralisação geral em 21 de maio, caso o governo não anuncie medidas efetivas sobre a política de preços do óleo diesel e de fiscalização rigorosa do piso mínimo do frete, assegura o caminhoneiro Wanderlei Alves, o Dedéco, de Curitiba (PR). "Se o diesel aumentar um centavo que seja e não houver efetiva fiscalização da aplicação do piso, a gente para no dia 21, quando a greve do ano passado completará um ano", garante.

3.2.2.5 Detecção de Idiomas

Para os experimentos envolvendo agrupamentos de textos de idiomas, 8 (oito) fontes de dados foram preparadas visando testes com o protótipo proposto na tese através da variação, na fonte de dados, da:

- quantidade de objetos total;
- quantidade de classes;
- quantidade de objetos por classe.

Foram utilizadas bases de dados texto e os objetos rotulados pelo seu idioma correspondente. Este problema de agrupamento foi escolhido devido sua boa distinção entre os objetos de dados sendo bem separado pelos algoritmos da literatura. As fontes 1 a 8 são de cunho pedagógico visando a demostração de experimentos no próximo capítulo.

Tabela 9 – Descrição das fontes de dados

Fontes de dados				
Fonte	Tipo	Descrição fonte	N de objetos	Expectativa
1	Não estruturado	Arquivos texto de 2 (duas) classes de linguagem.	55	Arquivos separados em 2 (duas) classes: Francês e Alemão.
2	Não estruturado	Arquivos texto de 5 (cinco) classes de linguagem.	15	Arquivos separados em 5 (cinco) classes: Francês, Alemão,Inglês,Português e Italiano.
3	Não estruturado	Arquivos texto de 3 (três) classes de linguagem.	30	Arquivos separados em 3 (três) classes: Francês, Alemão e Português.
4	Não estruturado	Arquivos texto de 4 (quatro) classes de linguagem. Característica particular do experimento: Número variado de objetos por classe.	38	Arquivos separados em 4 (quatro) classes: Francês, Alemão, Português e Italiano.
5	Não estruturado	Arquivos texto em português de 2 (duas) classes de linguagem.	8	Arquivos separados em de 2 (duas) classes: Política Brasileira e Política Internacional.
6	Não estruturado	Arquivos texto em português de 4 (quatro) classes de linguagem e 3 (três) subclasses para cada classe.	38	Arquivos separados em 4 (quatro)classes: Francês, Alemão, Italiano e Português e 3 (três) subclasses para cada classe: Biologia, Política Brasileira e Política Internacional.
7	Não estruturado	Arquivos texto de 4 (quatro) classes de linguagem.	37	Arquivos separados em 4 (quatro) classes: Francês, Alemão, Italiano e Português.
8	Não estruturado	Arquivos texto de 4 (quatro) classes de linguagem.	39	Arquivos separados em 4 (cinco) classes: Francês, Alemão, Italiano e Português.

Fonte: O próprio autor

3.2.3 Algoritmos

Utilizando a linguagem de programação Python em sua versão 3.7.2 aplicou-se algoritmos de compressão como LZW e Huffman ambos na etapa de *Codificação*, primeiro aplicado LZW e logo em seguida aplicado Huffman, com intuito de integrar e estender ao framework DAMICORE.

3.2.4 Controle de Versão

Ao longo de todo desenvolvimento serão testadas várias estratégias a fim analisar os resultados de cada uma e o sistema para controle de versão Git será utilizado de modo que não se perca nada dessa evolução do código e para que tenha um controle de toda alteração.

3.2.5 Visualização

Será utilizada uma árvore filogenética, ferramenta essa que tem suas origens da Biologia a fim de representar relações evolutivas entre organismos. Nesta pesquisa a árvore será construída através de uma matriz que representa a distância entre os objetos de dados, distância essa a qual é calculada com auxílio do algoritmo Neighbor Joining (NJ).

Desse modo, após a integração com o framework os resultados obtidos serão analisados, seguida a visualização será utilizado uma árvore filogenética, a ferramenta FigTree que auxilia nessa verificação visual. A montagem dos gráficos ficará a cargo da ferramenta Google Sheets.

4 RESULTADOS E DISCUSSÃO

4.0.1 Algoritmos adaptativos envolvendo objetos comprimidos do tipo texto

Os resultados serão analisados na execução de 8 (oito) experimentos envolvendo objetos de dados não-estruturados no formato texto, comprimidos e mensurados com NCD em agrupamento hierárquico.

A Tabela 10 apresenta os resultados dos experimentos obtidos da fonte de dados 1 utilizando NCD com textos dos idiomas francês e alemão comprimidos por uma MT e compondo um total de 55 objetos. Enquanto que para objetos com grânulos de informação configurados para a MT RLE e MT BWT obteve-se 98,11% de acurácia, para a MT LZW, as hipóteses H não apresentam nenhum erro na classificação, ou seja, $H = Ex$.

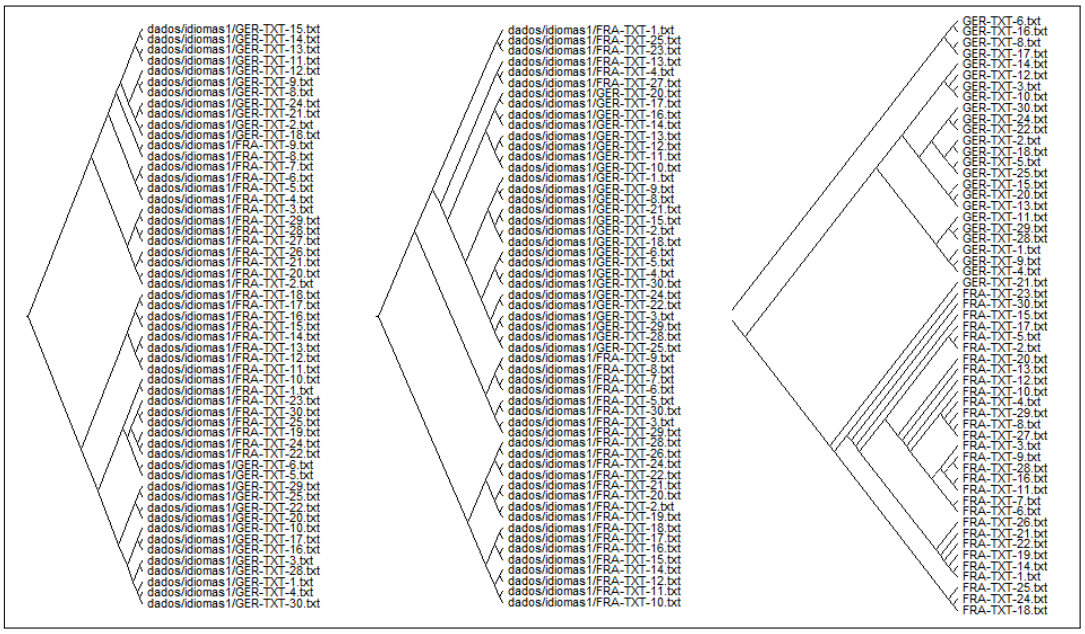
Tabela 10 – Resultados dos experimentos com a fonte de dados 1 com NCD.

Fonte de dados 1 **Dist. NCD - Gráfico:** $B.1.1$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	1	2	2
Acurácia	98,11%	98,11%	100%

Fonte: O próprio autor

Os resultados da Tabela 10 podem ser observados graficamente na Figura 4. Observa-se que os resultados utilizando a MT LZM separa-se de maneira mais assertiva as duas classes consideradas com todos os objetos agrupados de forma correta. Este resultado se mostrou presente em todos os experimentos envolvendo MT LZM realizados nas fontes de dados de número 1 a 8, sendo assim, esta configuração de dependência da MT será considerada como limite máximo ou ideal de comparação das hipóteses H.

Figura 4 – Cladograma de 55 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 11 apresenta os resultados dos experimentos obtidos da fonte de dados 2 utilizando NCD com textos das línguas francês, alemão, inglês, português e italiano comprimidos por uma MT e compondo um total de 15 objetos. Enquanto que para objetos com grânulos de informação configurados para a MT RLE obteve-se 90% de acurácia, para a MT BWT obteve-se 80%. Hipóteses H com MT LZW seguem em acurácia de 100% em todos os experimentos como já mencionado.

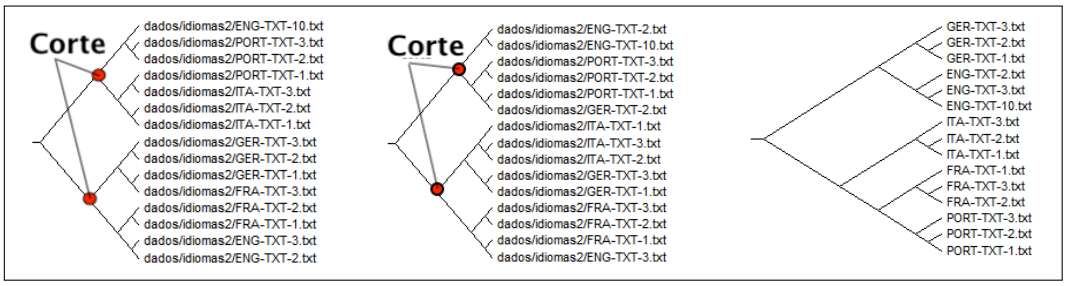
Tabela 11 – Resultados dos experimentos com a fonte de dados 2 com NCD.

Fonte de dados 2 **Dist. NCD - Gráfico:** $B.1.2$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	2	3	5
Acurácia	90%	80%	100%

Fonte: O próprio autor

Os resultados da Tabela 11 podem ser observados graficamente na Figura 5.

Figura 5 – Cladograma de 15 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 12 apresenta os resultados dos experimentos obtidos da fonte de dados 3 utilizando NCD com textos das línguas francês, alemão e português comprimidos por uma MT e compondo um total de 30 objetos. Enquanto que para objetos com grânulos de informação configurados para a MT RLE obteve-se 100% de acurácia, para a MT BWT obteve-se 88,89%.

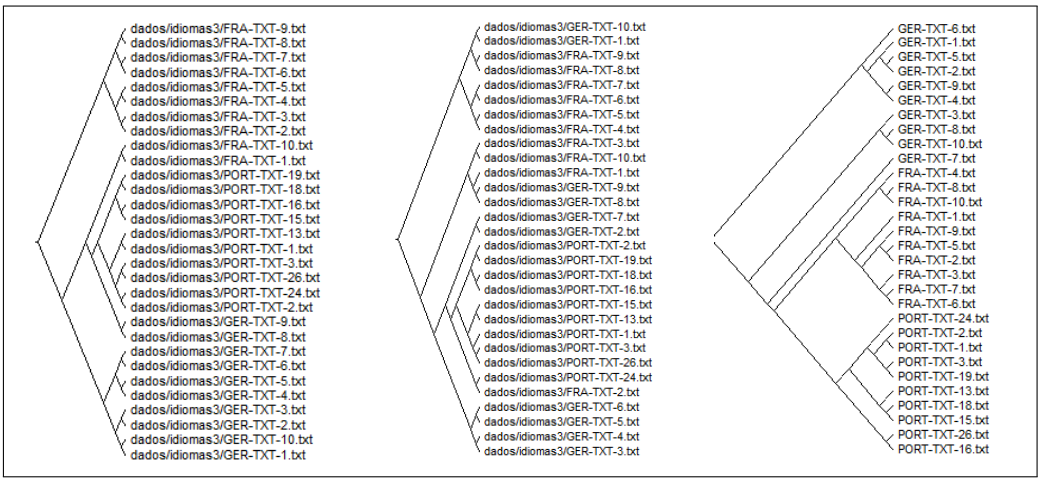
Tabela 12 – Resultados dos experimentos com a fonte de dados 3 com NCD.

Fonte de dados 3 **Dist. NCD - Gráfico:** $B.1.3$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	3	1	3
Acurácia	100%	88,89%	100%

Fonte: O próprio autor

Os resultados da Tabela 12 podem ser observados graficamente na Figura 6.

Figura 6 – Cladograma de 30 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 13 apresenta os resultados dos experimentos obtidos da fonte de dados 4 utilizando NCD com textos das línguas francês, alemão, português e italiano comprimidos por uma MT e compondo um total de 38 objetos. Enquanto que para objetos com grânulos de informação configurados para a MT RLE obteve-se 85,29% de acurácia, para a MT BWT obteve-se 85,29%.

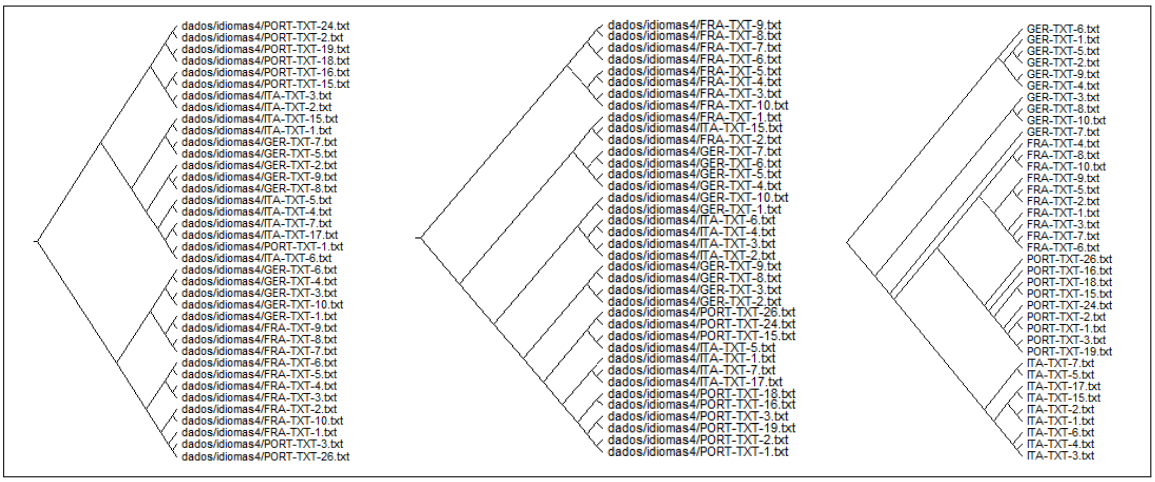
Tabela 13 – Resultados dos experimentos com a fonte de dados 4 com NCD.

Fonte de dados 4 **Dist. NCD - Gráfico:** $B.1.4$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	1	0	4
Acurácia	85,29%	85,29%	100%

Fonte: O próprio autor

Os resultados da Tabela 13 podem ser observados graficamente na Figura 7.

Figura 7 – Cladograma de 38 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 14 apresenta os resultados dos experimentos obtidos da fonte de dados 5 utilizando NCD com textos em português de política brasileira e política internacional comprimidos por uma MT e compondo um total de 8 objetos. Para objetos com grânulos de informação configurados para a MT com RLE e BWT obteve-se 100%.

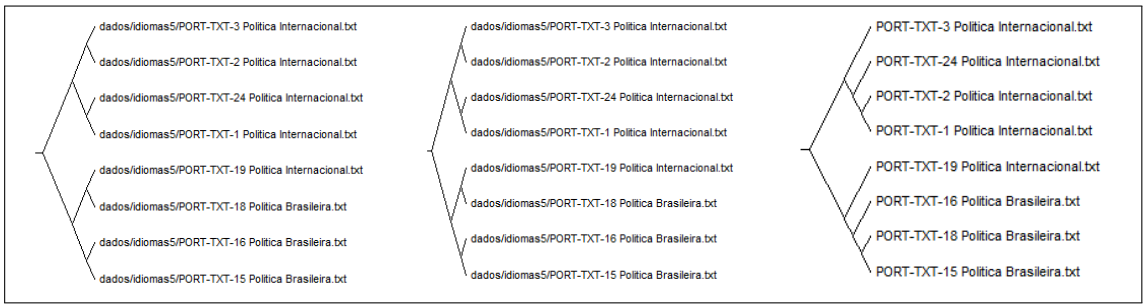
Tabela 14 – Resultados dos experimentos com a fonte de dados 5 com NCD.

Fonte de dados 5 **Dist. NCD - Gráfico:** $B.1.5$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	1	1	2
Acurácia	100%	100%	100%

Fonte: O próprio autor

Os resultados da Tabela 14 podem ser observados graficamente na Figura 8.

Figura 8 – Cladograma de 8 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 15 apresenta os resultados dos experimentos obtidos da fonte de dados 6 utilizando NCD com textos em francês, alemão, italiano, e português (subdividido nas classes Biologia, Política Brasileira e Política Internacional) comprimidos por uma MT e compondo um total de 38 objetos. Para objetos com grânulos de informação configurados para a MT RLE obteve-se 82,35% enquanto que para a MT BWT obteve-se 85,29%.

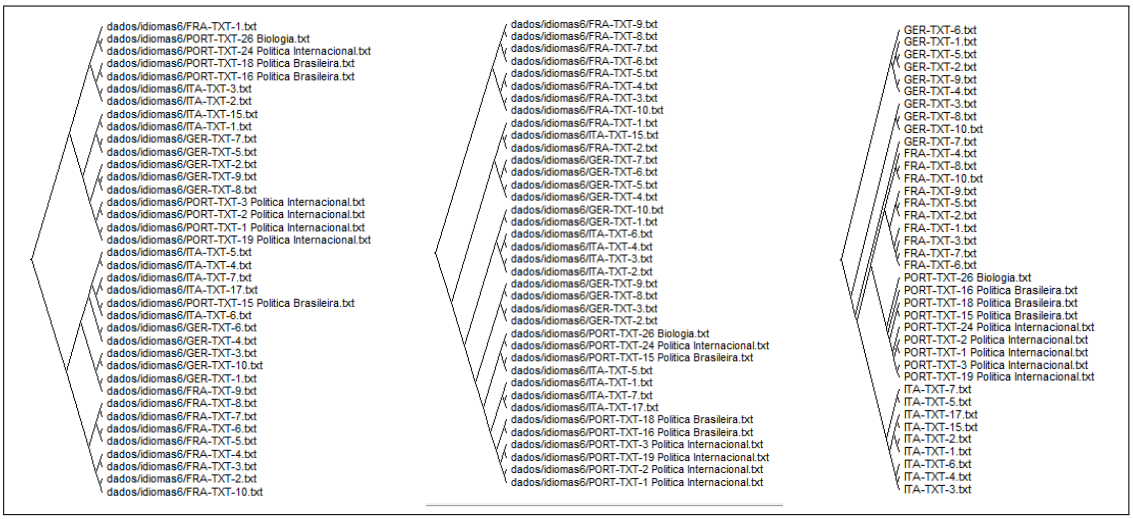
Tabela 15 – Resultados dos experimentos com a fonte de dados 6 com NCD.

Fonte de dados 6 **Dist. NCD - Gráfico:** $B.1.6$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	0	1	4
Subclasses completamente identificadas	0	1	3
Acurácia no agrupamento das classes	82,35%	85,29%	100%
Acurácia no agrupamento das subclasses	66,67%	83,33%	100%

Fonte: O próprio autor

Os resultados da Tabela 15 podem ser observados graficamente na Figura 9.

Figura 9 – Cladograma de 38 objetos comprimidos por MT RLE (Agrup. à esquerda), BWT (Agrup. do meio) e LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 16 apresenta os resultados dos experimentos obtidos da fonte de dados 7 utilizando NCD com textos em francês, alemão, italiano, e português (subdividido nas classes Política Brasileira e Política Internacional) comprimidos por uma MT e compondo um total de 37 objetos. Para objetos com grânulos de informação configurados para a MT RLE obteve-se 90,91% enquanto que para a MT BWT obteve-se 87,88%.

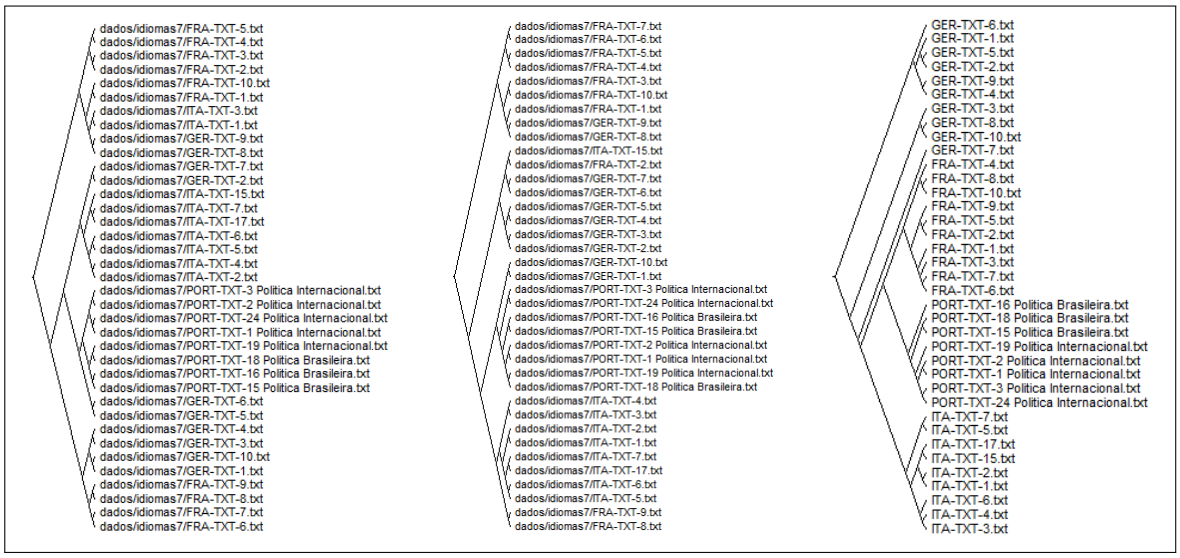
Tabela 16 – Resultados dos experimentos com a fonte de dados 7 com NCD.

Fonte de dados 7 **Dist. NCD - Gráfico:** $B.1.7$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	1	1	4
Subclasses completamente identificadas	2	0	2
Acurácia no agrupamento das classes	90,91%	87,88%	100%
Acurácia no agrupamento das subclasses	100%	66,67%	100%

Fonte: O próprio autor

Os resultados da Tabela 16 podem ser observados graficamente na Figura 10.

Figura 10 – Cladograma de 37 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



Fonte: O próprio autor

A Tabela 17 apresenta os resultados dos experimentos obtidos da fonte de dados 8 utilizando NCD com textos em francês, alemão, italiano, e português (subdividido nas classes Saúde, Política Brasileira e Política Internacional) comprimidos por uma MT e compondo um total de 39 objetos. Para objetos com grânulos de informação configurados para a MT RLE obteve-se 88,57% enquanto que para a MT BWT obteve-se 100%.

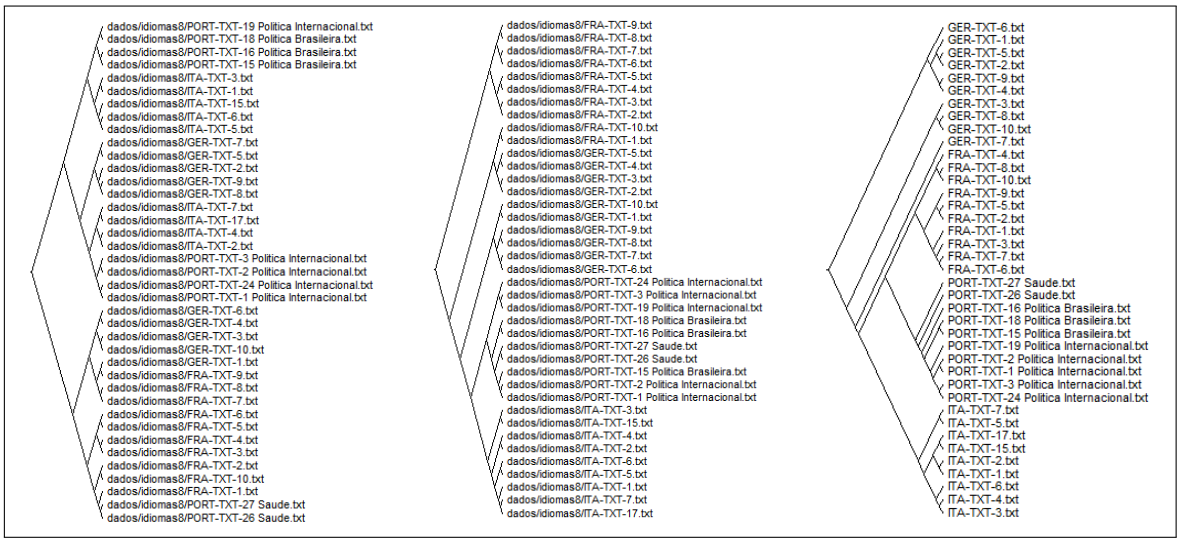
Tabela 17 – Resultados dos experimentos com a fonte de dados 8 com NCD.

Fonte de dados 8 **Dist. NCD - Gráfico:** $B.1.8$			
MT	**RLE**	**BWT**	**LZW**
Informação: (Resolução/Semântica)	símbolo	digrama	string
Classes completamente identificadas	1	4	4
Subclasses completamente identificadas	1	1	3
Acurácia no agrupamento das classes	88,57%	100%	100%
Acurácia no agrupamento das subclasses	85,71%	71,43%	100%

Fonte: O próprio autor

Os resultados da Tabela 17 podem ser observados graficamente na Figura 11.

Figura 11 – Cladograma de 39 objetos comprimidos por MT RLE (Agrup. à esquerda), MT BWT (Agrup. do meio) e MT LZW (Agrup. à direita)



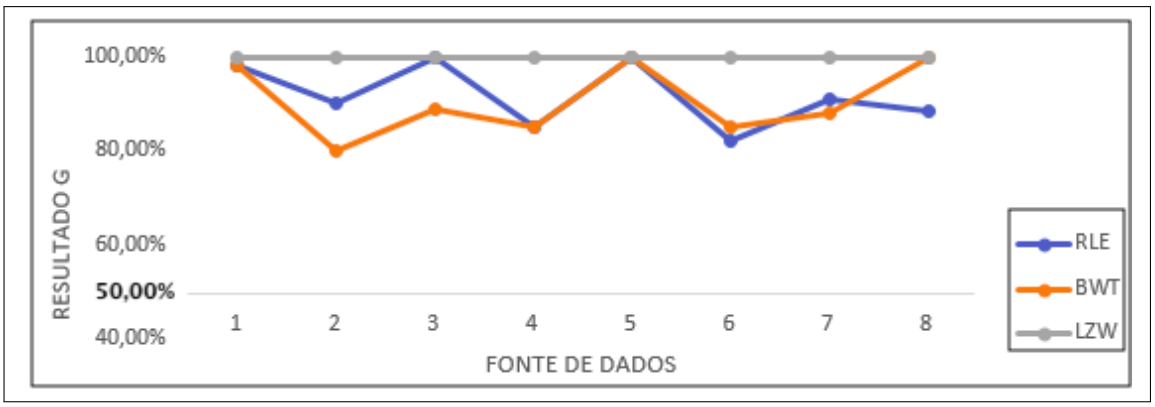
Fonte: O próprio autor

4.1 Resultados empíricos

Diante dos experimentos realizados foram montados dois gráficos para apresentar a acurácia dos agrupamentos de classes e subclasses baseado no coeficiente G em função de cada uma das fontes de dados.

O resultado obtido no agrupamento de classes com 5 (cinco) idiomas distribuídos em 8 (oito) fonte de dados é ilustrado no gráfico da Figura 12

Figura 12 – Resultado G do agrupamento de classes (idiomas)



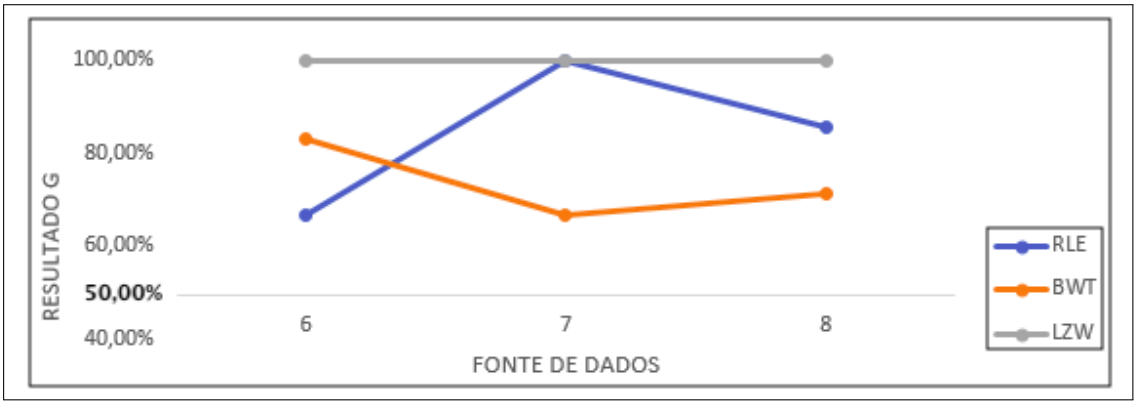
Fonte: O próprio autor

Conforme ilustra a Tabela 9 as fontes de dados possuem diferentes características, simulando diversas situações. Há fonte com variados números de objetos assim como com

diferentes números de classes. Mesmo com esse cenário o algoritmo MT LZW obteve uma acurácia de 100% em todas as fontes de dados, isto é, obteve uma média de 100% nos resultados e apresentado assim um desvio padrão igual a 0. Os algoritmos MT RLE e MT BWT apresentaram algumas variações nos resultados, o RLE obteve uma média igual a 91,90% e uma variância de 6,77 enquanto o BWT obteve uma média de 90,68% e variância de 7,67. Com esses resultados fica comprovado que o LZW está estatisticamente à frente dos demais se tratando do agrupamento de classes.

A Figura 13 ilustra os resultados obtidos no agrupamento de subclasses dentro do idioma português, somente as fontes de dados 6, 7 e 8 apresentavam conjuntos de dados com subclasses.

Figura 13 – Resultado G do agrupamento de subclasses dentro da classe idiomas



Fonte: O próprio autor

Após análise gráfica e estatística dos resultados nota-se que o algoritmo MT LZW mantém assertividade em 100% apresentando assim uma média na acurácia de 100% com desvio padrão igual 0. O algoritmo MT RLE demostra algumas variações e obtém uma média de 84,13% na acurácia e desvio padrão de 16,72 enquanto o MT BWT possui uma média de 73,81% na acurácia e um desvio padrão de 8,58. Nos resultados com agrupamento de subclasses o algoritmo LZW também mantém melhores resultados justificando assim sua escolha para a realização dos demais experimentos com conjuntos de dados variados.

4.2 Resultados em conjuntos de dados texto com semântica

Para realização dos experimentos em conjuntos de dados texto com semântica, os dados foram separados em bases, cada base sendo representada por uma tabela, as bases ainda dispõem de fontes com quantidades distintas de classes.

A Tabela 18 apresenta em cada linha uma fonte distinta, incrementando sempre em sua quantidade com o intuito de analisar o comportamento do algoritmo em diversos cenários.

Tabela 18 – Bases de dados com sinopses de filmes e divisão por classes

Num. de CLASSES	**Sinopses de filmes**
2	AÇÃO(9), BIOGRAFIA(7)
3	AÇÃO(9), BIOGRAFIA(7), COMÉDIA(5)
4	AÇÃO(9), BIOGRAFIA(7), COMÉDIA(5), DRAMA(3)
5	AÇÃO(9), BIOGRAFIA(7), COMÉDIA(5), DRAMA(3), HISTÓRIA(4)
6	AÇÃO(9), BIOGRAFIA(7), COMÉDIA(5), DRAMA(3), HISTÓRIA(4), ROMANCE(6)
7	AÇÃO(9), BIOGRAFIA(7), COMÉDIA(5), DRAMA(3), HISTÓRIA(4), ROMANCE(6), GUERRA(6)

Fonte: O próprio autor

A Tabela 19 traz a descrição das 4 (quatro) fontes de dados utilizadas, simulando um ambiente com as classes todas balanceadas e variando de 2 (duas) a 5 (cinco) classes.

Tabela 19 – Bases de dados com abstracts de artigos e divisão por classes

Num. de CLASSES	**Abstracts de artigos**
2	BIOLOGIA(50), QUÍMICA(50)
3	BIOLOGIA(50), QUÍMICA(50), CIÊNCIA DA COMPUTAÇÃO(50)
4	BIOLOGIA(50), QUÍMICA(50), CIÊNCIA DA COMPUTAÇÃO(50), FÍSICA(50)
5	BIOLOGIA(50), QUÍMICA(50), CIÊNCIA DA COMPUTAÇÃO(50), FÍSICA(50), CIÊNCIAS SOCIAIS(50)

Fonte: O próprio autor

A Tabela 20 dispõe de uma fonte de dados com 2 (duas) classes balanceadas contendo 100 objetos cada uma dessas classes, simulando assim um cenário onde envolve uma situação binária.

Tabela 20 – Bases de dados com avaliação de produtos e divisão por classes

Num. de CLASSES	**Avaliação de produtos**
2	NEGATIVO(100), POSITIVO(100)

Fonte: O próprio autor

A Tabela 21 contem um total de 6 classes distribuídas em 5 classes. As classes foram bem balanceadas onde cada uma contava com 50 objetos.

Tabela 21 – Bases de dados com textos contendo sentimentos e divisão por classes

Num. de CLASSES	**Texto com sentimentos**
2	TRISTEZA(50), SURPRESA(50)
3	TRISTEZA(50), SURPRESA(50), ALEGRIA(50)
4	TRISTEZA(50), SURPRESA(50), ALEGRIA(50), DESGOSTO(50)
5	TRISTEZA(50), SURPRESA(50), ALEGRIA(50), DESGOSTO(50), RAIVA(50)
6	TRISTEZA(50), SURPRESA(50), ALEGRIA(50), DESGOSTO(50), RAIVA(50), MEDO(50)

Fonte: O próprio autor

A Tabela 22 representa uma base de dados com 3 (três) classes distribuídas em 2 (duas) fontes em cenário onde os dados são balanceados contendo aqui 100 objetos cada classe.

Tabela 22 – Bases de dados com notícias do Twitter e divisão por classes

Num. de CLASSES	**Twitter - notícias**
2	NEGATIVO(100), NEUTRO(100)
3	NEGATIVO(100), POSITIVO(100), NEUTRO(100)

Fonte: O próprio autor

A Tabela 23 representa uma base de dados com 5 classes distribuídas entre 4 fontes, as fontes possuem quantidades variadas simulando um cenário onde os dados não estão balanceados.

Tabela 23 – Bases de dados com artigos noticiários e divisão por classes

Num. de CLASSES	**Artigos - noticiários**
2	ECONOMIA(15), ESPORTE(21)
3	ECONOMIA(15), ESPORTE(21), POLÍTICA(17)
4	ECONOMIA(15), ESPORTE(21), POLÍTICA(17), SAÚDE(27)
5	ECONOMIA(15), ESPORTE(21), POLÍTICA(17), SAÚDE(27), TECNOLOGIA(27)

Fonte: O próprio autor

A Tabela 24 representa uma base de dados com 5 (cinco) idiomas separados em 4 (quatro) fontes, as fontes representam um cenário onde os dados não estão balanceados,

variando a quantidade de objetos em cada classe.

Tabela 24 – Bases de dados com idiomas e divisão por classes

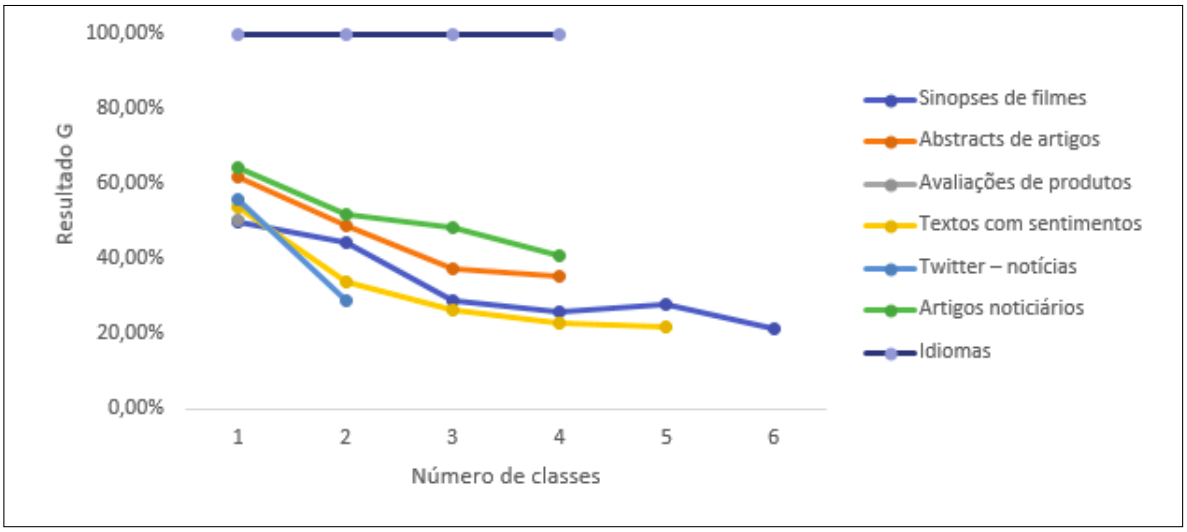
Num. de CLASSES	**Idiomas**
2	FRANCÊS(30), ALEMÃO(30)
3	FRANCÊS(10), ALEMÃO(10), PORTUGUÊS(10)
4	FRANCÊS(10), ALEMÃO(10), PORTUGUÊS(9), ITALIANO(9)
5	FRANCÊS(3), ALEMÃO(3), PORTUGUÊS(3), ITALIANO(3), INGLÊS(3)

Fonte: O próprio autor

De tal modo, após colocar o algoritmo para gerar os resultados sob as bases de dados coletadas, é importante salientar que foram utilizadas 4 estratégias diferentes para obtenção dos resultados.

Na primeira bateria de experimentos é utilizada a estratégia que envolve a implementação do algoritmo LZW de forma tradicional. O resultado obtido é apresentado na Figura 14. Todos os experimentos aplicam a codificação de Huffman, devido a isso não se faz necessário mencioná-lo nos experimentos seguintes.

Figura 14 – Resultados da estratégia 1



Fonte: O próprio autor

Tabela 25 – Resultado G da primeira bateria de testes

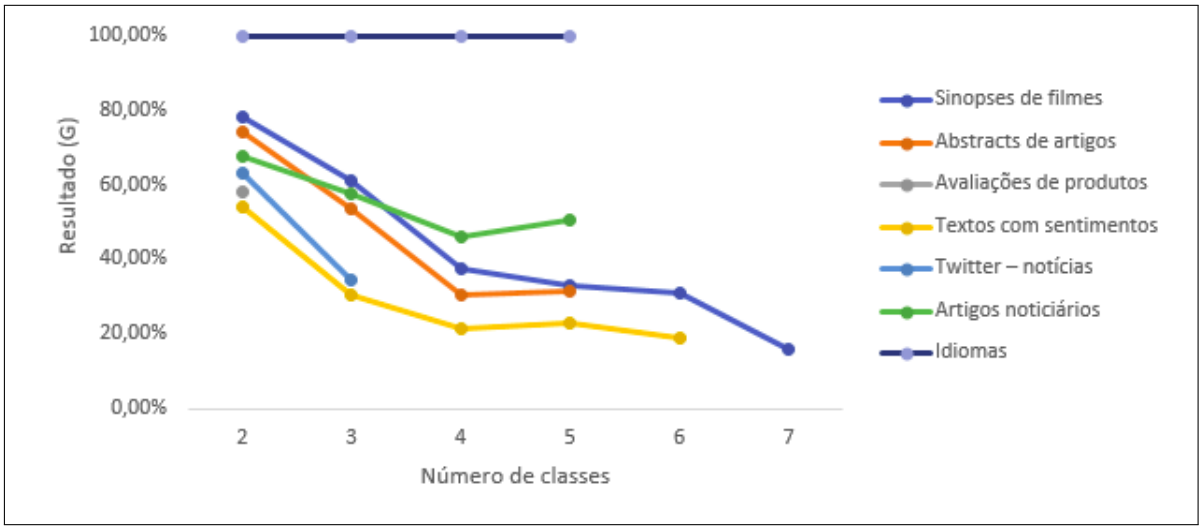
	2 classes	**3**	**4**	**5**	**6**	**7 classes**
Sinopses de filmes	50,00%	44,44%	29,17%	25,93%	28,12%	21,62%
Abstracts de artigos	62,24%	48,98%	37,76%	35,51%	-	-
Avaliação de produtos	50,51%	-	-	-	-	-
Texto com sentimentos	54,08%	34,01%	26,53%	22,86%	22,11%	
Twitter - notícias	56,06%	28,96%	-	-	-	-
Artigos noticiários	64,71%	52,00%	48,68%	41,18%	-	-
Idiomas	100,00%	100,00%	100,00%	100,00%	-	-

Fonte: O próprio autor

Como visto no Capítulo 3 de Materiais e Métodos a base composta por sinopses de filmes onde a fonte composta por duas classes, o resultado foi de 50% de acerto, mesma taxa de acerto de um algoritmo aleatório (isso nestes experimentos são considerados como resultados ruins). A base que é composta por notícias do Twitter também obteve resultados ruins na fonte 2, perdendo para um algoritmo aleatório. Portanto, percebe-se que a estratégia de usar somente o LZW tem resultados abaixo do esperado, pois são encontrados poucos padrões e os resultados ficam próximos de algoritmos aleatórios. É notório que conforme a quantidade de classes aumentam a taxa de acerto diminui. Um fator que causa esse resultado é a menor exigência de um processamento semântico na identificação de idiomas.

A segunda estratégia adotada é um único dicionário para todo o *corpus* de documentos, diferente da estratégia tradicional como na primeira, a unidade mínima considerada para codificação deixa de ser um caractere para ser uma palavra. O resultado dessa bateria de experimentos é mostrado na Figura 15.

Figura 15 – Resultados da estratégia 2



Fonte: O próprio autor

Tabela 26 – Resultado G da segunda bateria de testes

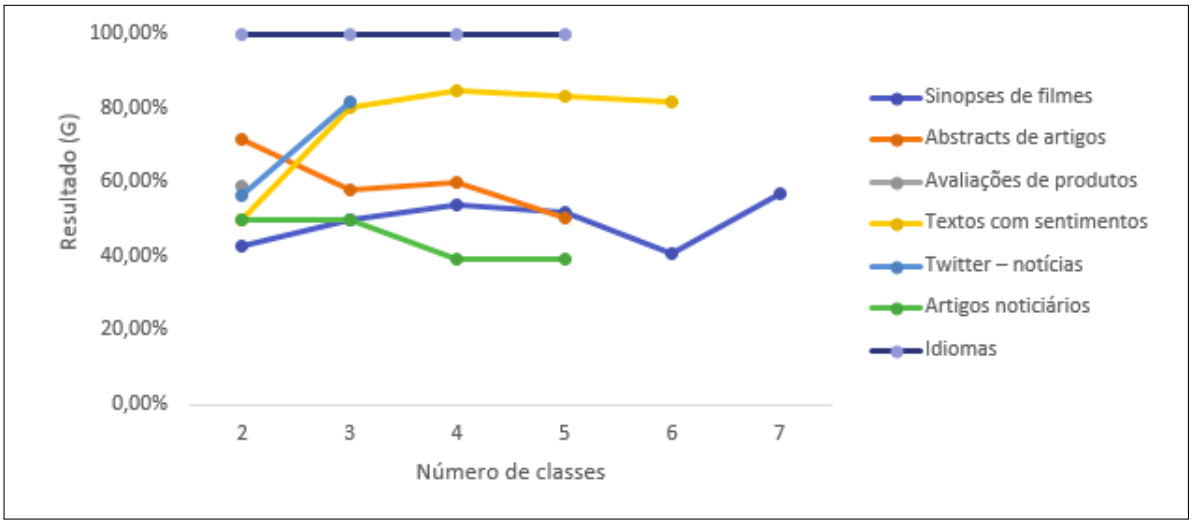
	2 classes	**3**	**4**	**5**	**6**	**7 classes**
Sinopses de filmes	78,57%	61,11%	37,50%	33,33%	31,25%	16,22%
Abstracts de artigos	74,49%	53,74%	30,61%	31,43%	-	-
Avaliação de produtos	58,08%	-	-	-	-	-
Texto com sentimentos	54,08%	30,61%	21,43%	23,27%	19,05%	
Twitter - notícias	63,13%	34,68%	-	-	-	-
Artigos noticiários	67,65%	58,00%	46,05%	50,98%	-	-
Idiomas	100,00%	100,00%	100,00%	100,00%	-	-

Fonte: O próprio autor

Diante dos resultados da bateria 2 de experimentos nota-se uma sutil melhora nos resultados, mas conforme aumenta a quantidade de classes os resultados ainda caem consideravelmente, em casos específicos como as fontes com 3 e 4 classes da base que é composta por texto com sentimentos, base essa que é composta por tweets que expressam emoções, ainda tem resultados inferiores à de algoritmos aleatórios. Assim como na bateria 1, aqui o algoritmo tem dificuldade em encontrar padrões em bases compostas por arquivos com poucos caracteres, como um *tweet* que neste contexto é considerado um texto pequeno. Um *tweet* até Outubro de 2017 continha um limite máximo de 140 caracteres, a partir de Novembro de 2017 esse limite dobrou de tamanho passando a ser 280.

Em busca de contornar os problemas de encontrar padrões em textos pequenos a estratégia 3 adota um método de pré-processamento textual para remover *stopwords*, diminuindo ainda mais os textos pequenos, mas mantendo somente padrões relevantes para o significado da expressão. A Figura 16 ilustra o resultado da terceira bateria de experimentos.

Figura 16 – Resultados da estratégia 3



Fonte: O próprio autor

Tabela 27 – Resultado G da terceira bateria de testes

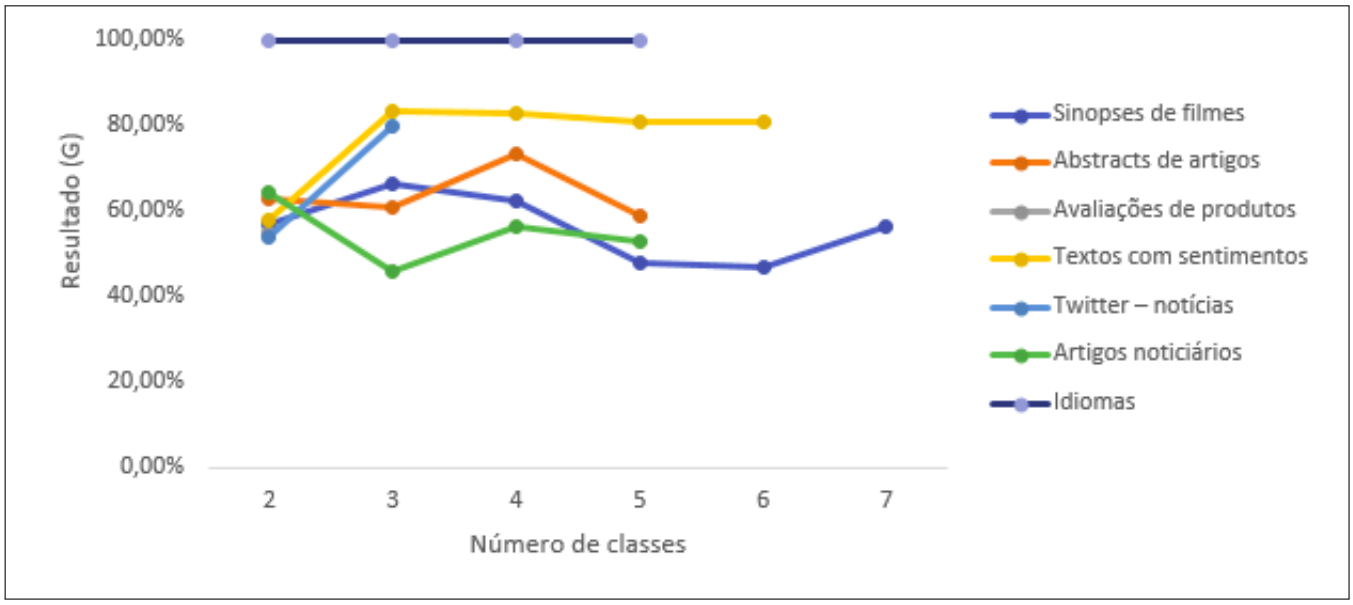
	2 classes	**3**	**4**	**5**	**6**	**7 classes**
Sinopses de filmes	42,86%	50,00%	54,17%	51,85%	40,62%	56,76%
Abstracts de artigos	71,43%	57,82%	60,20%	50,20%	-	-
Avaliação de produtos	59,09%	-	-	-	-	-
Texto com sentimentos	50,00%	80,27%	84,69%	83,27%	81,63%	
Twitter - notícias	56,57%	81,82%	-	-	-	-
Artigos noticiários	50,00%	50,00%	39,47%	39,22%	-	-
Idiomas	100,00%	100,00%	100,00%	100,00%	-	-

Fonte: O próprio autor

Com essa estratégia nota-se que conforme a quantidade de classes aumenta a taxa de acerto não cai tão drasticamente como nas baterias anteriores, em alguns casos como na base composta por textos com sentimentos e Twitter com notícias houve melhora significativa com o aumento de classes. As demais bases mantém uma uniformidade, bases essas compostas pelos textos com maior quantidade de características no contexto deste experimento.

Por conseguinte a quarta estratégia aplicada, estende e melhora a estratégia anterior, implementando tag de negação, combinação com bigramas, aplicando a remoção das *stopwords* e considerando a frequência com que as palavras que ainda restaram aparecem no documento de texto, todas essas técnicas de pré-processamento discutidas no Capítulo 2. Após executar o algoritmo os resultados obtidos são mostrados na Figura 17.

Figura 17 – Resultados da estratégia 4



Fonte: O próprio autor

Tabela 28 – Resultado G da quarta bateria de testes

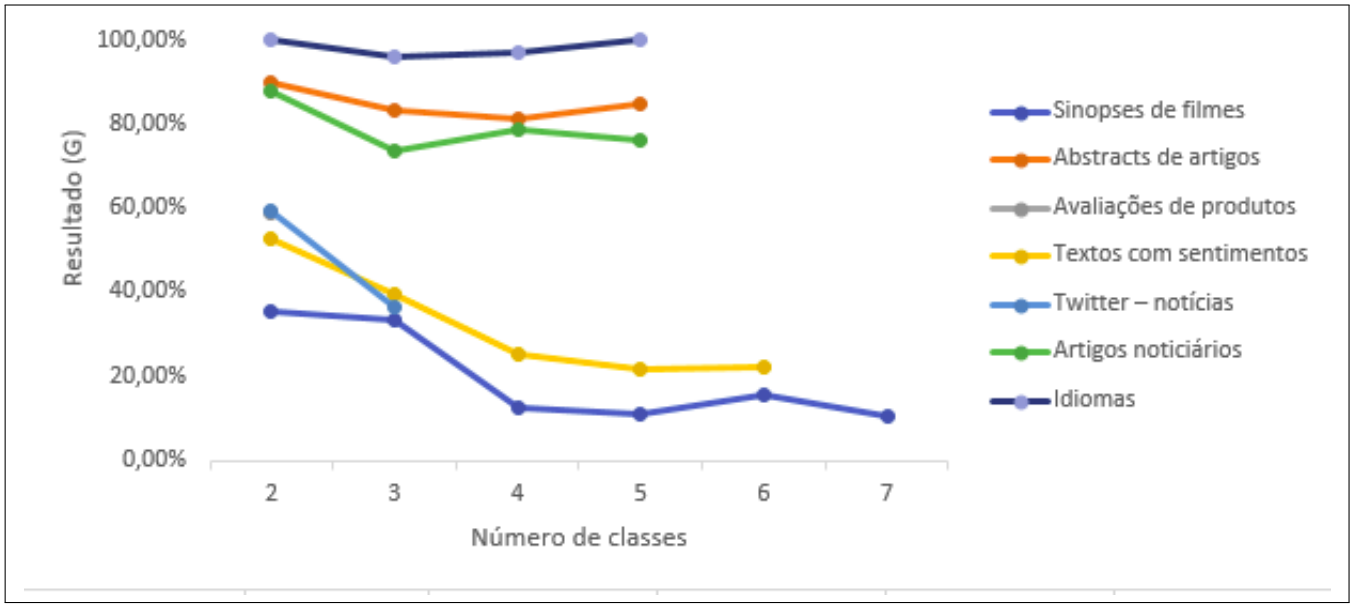
	2 classes	**3**	**4**	**5**	**6**	**7 classes**
Sinopses de filmes	71,43%	66,67%	75,00%	51,85%	43,75%	51,35%
Abstracts de artigos	70,41%	67,35%	74,49%	71,02%	-	-
Avaliação de produtos	61,11%	-	-	-	-	-
Texto com sentimentos	61,22%	81,63%	85,71%	84,49%	82,65%	
Twitter - notícias	60,61%	82,15%	-	-	-	-
Artigos noticiários	52,94%	62,00%	52,63%	60,78%	-	-
Idiomas	100,00%	100,00%	100,00%	100,00%	-	-

Fonte: O próprio autor

Nesta quarta bateria de experimentos os resultados bons da terceira bateria foram mantidos e os regulares foram melhorados, então com essa estratégia foram solucionados problemas de modo que não há mais casos onde a taxa de acertos é menor que a de um algoritmo aleatório e o aumento da quantidade de classes deixa de diminuir drasticamente a taxa de acerto, assim como os textos com poucos caracteres agora mostram resultados expressantes, dentro deste contexto os objetos de dados com até 140 caracteres são considerados textos com poucos caracteres.

Para fins comparativos foi executado o ambiente DAMICORE executando as mesmas bases de dados e o mesmo coeficiente, o resultado é mostrado na Figura 18 e a Tabela 29 detalha com exatidão a taxa de acerto para cada teste.

Figura 18 – Resultados DAMICORE



Fonte: O próprio autor

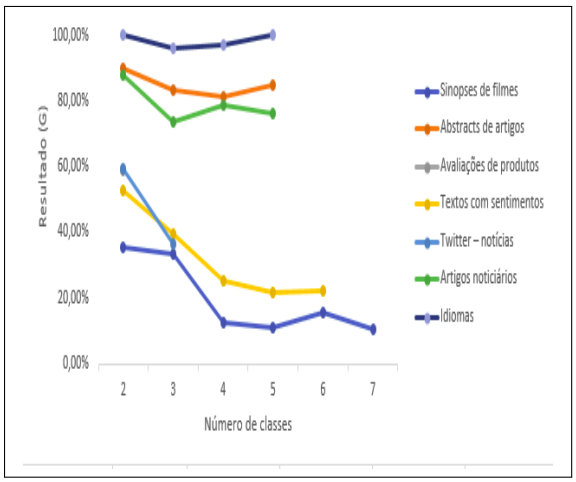
Tabela 29 – Resultado G dos testes DAMICORE

	2 classes	**3**	**4**	**5**	**6**	**7 classes**
Sinopses de filmes	35,71%	33,33%	12,50%	11,11%	15,62%	10,81%
Abstracts de artigos	89,98%	83,67%	81,63%	84,90%	-	-
Avaliação de produtos	59,09%	-	-	-	-	-
Texto com sentimentos	53,06%	39,46%	25,51%	21,63%	22,45%	
Twitter - notícias	59,60%	36,70%	-	-	-	-
Artigos noticiários	88,24%	74,00%	78,95%	76,47%	-	-
Idiomas	100,00%	96,30%	97,06%	100,00%	-	-

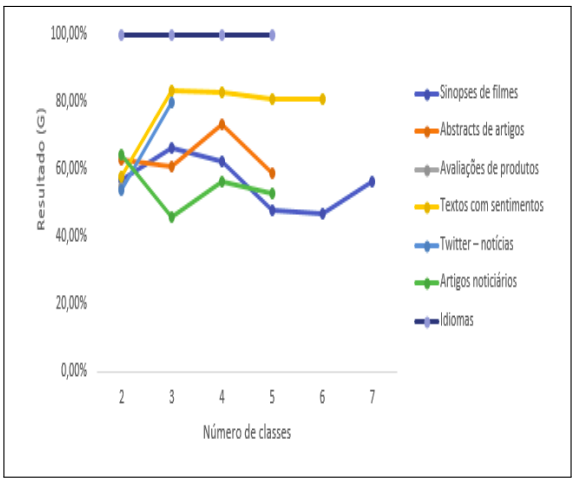
Fonte: O próprio autor

As Figuras 19 e 20 dispõem de um comparativo entre a solução proposta e o DAMICORE.

Figura 19 – Comparativo geral DAMICORE X Solução Proposta

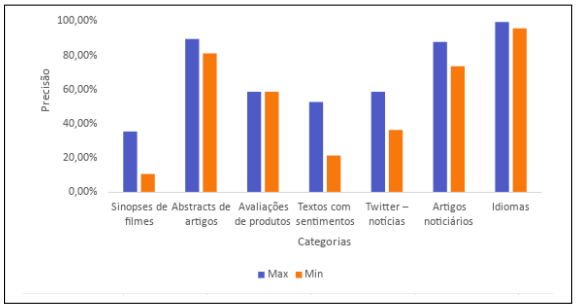


(a) Resultado DAMICORE

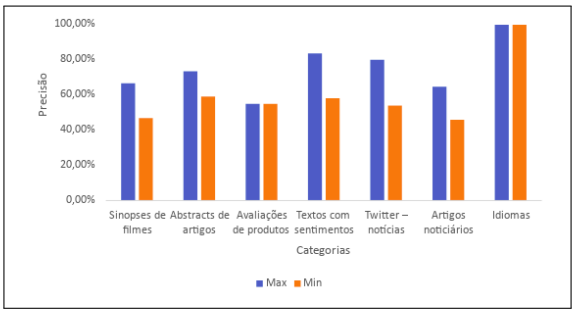


(b) Resultado Solução Proposta

Figura 20 – Comparativo melhores e piores resultados DAMICORE X Solução Proposta



(a) Melhores e piores resultados DAMICORE



(b) Melhores e piores resultados Solução Proposta

Fonte: O próprio autor

Desta forma, nota-se que no DAMICORE textos com pouca quantidade de caracteres, obtêm resultados de baixa precisão, isto ocorre nas bases de dados: “Twitter -

notícias", "Avaliação de produtos", "Texto com sentimentos" e "Sinopses de filmes", devido ao fato de textos com essas caraterísticas repetir poucos padrões e o foco ficar mais no conteúdo semântico. A solução proposta explora técnicas para pré-processamento textual que obtêm melhores resultados nesses tipos de ocorrência, ocorrências que caracterizam análise de sentimentos, exceto "Sinopses de filmes".

5 CONCLUSÃO

Diante a pesquisa realizada, o presente estudo possibilitou o desenvolvimento de um software feito para agrupar dados em formato de texto de acordo com sua categoria usando de algoritmos de compressão, reconhecimento de padrões juntamente com técnicas de Processamento de Linguagem Natural para interpretar o conteúdo textual, gerando como saída os textos agrupados em uma árvore filogenética.

5.1 Considerações Finais

Em suma, observa-se que os resultados iniciais demostraram que ajustes realizados nos algoritmos de compressão dos compactadores disponíveis no mercado associados a divisão da abordagem de compressão em *Granulação* e *Codificação* geram representações mais próximas das expectativas de contextos semânticos exigidos em cenários de RP.

5.2 Trabalhos Futuros

Com os ótimos resultados obtidos na detecção de idiomas, planeja-se o uso dessa saída gerada para traduzir todo texto para o idioma inglês e melhorar o agrupamento de textos que exigem mais da semântica do textual. Em hipótese, esse desenvolvimento e prática atualizada facilitará o tratamento, visto que estará centralizado em um único idioma.

Com o propósito de aprimorar os resultados, também é planejado o uso de um algoritmo específico para detecção de sinônimos, fazendo com que palavras distintas e com mesmo significado apontem para uma mesma referência no dicionário.

Referências Bibliográficas

ARAUJO, V. M. R. H. de. Sistemas de informação: nova abordagem teórico-conceitual. **Instituto Brasileiro de Informação, Ciência e Tecnologia**, v. 22, 1995. Citado na página 7.

BORDIGNON, A. C. de A. et al. Natural language processing in business process identification and modeling: A systematic literature review. In: **Proceedings of the XIV Brazilian Symposium on Information Systems**. New York, NY, USA: ACM, 2018. (SBSI'18), p. 25:1–25:8. ISBN 978-1-4503-6559-8. Disponível em: <http://doi.acm.org/10.1145/3229345.3229373>. Citado na página 6.

BRAZIL, A. L. **O algoritmo LZW**. 20–. Disponível em: <http://www.ic.uff.br/~aconci/LZW.pdf>. Acesso em: 07/12/2019. Citado na página 11.

BRITO, E. M. N. **Mineração de Textos: Detecção automática de sentimentos em comentários nas mídias sociais**. Universidade FUMEC: Faculdade de Ciências empresariais, 2016. Citado 2 vezes nas páginas 3 e 9.

CAMPOS, T. E. D. Técnicas de seleção de características com aplicações em reconhecimento de faces. Instituto de Matemática e Estatística, Universidade de São Paulo, 2001. Citado na página 12.

CASELI, H. Tradução automática: estratégias e limitações. v. 11, p. 1782, 12 2017. Citado na página 7.

CUNHA, J. P. A. P. da. **Compressão de dados baseada em Dicionários utilizando o algoritmo LZW**. 20–. Disponível em: <http://seer.pucgoias.edu.br/files/journals/3/articles/518/submission/review/518-1798-1-RV.doc>. Acesso em: 07/12/2019. Citado na página 11.

GONÇALVES, T. et al. Analysing part-of-speech for portuguese text classification. In: GELBUKH, A. (Ed.). **Computational Linguistics and Intelligent Text Processing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 551–562. ISBN 978-3-540-32206-1. Citado 2 vezes nas páginas 5 e 8.

HUANG, J.; ZHOU, M.; YANG, D. Extracting chatbot knowledge from online discussion forums. In: **Proceedings of the 20th International Joint Conference on Artifical Intelligence**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007. (IJCAI'07), p. 423–428. Disponível em: <http://dl.acm.org/citation.cfm?id=1625275.1625342>. Citado na página 7.

JACKSON, P.; MOULINIER, I. **Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization**. [S.l.]: John Benjamins Pub., 2007. (Natural language processing). ISBN 9789027249920. Citado na página 8.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Citado na página 9.

MONTEIRO, A. et al. Algoritmos para reconhecimento de padrões. 2008. Citado 2 vezes nas páginas 12 e 13.

MONTEIRO, S. D. et al. Sistemas de recuperação da informação e o conceito de relevância nos mecanismos de busca: semântica e significação. **Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação**, v. 22, p. 161–175, 9 2017. Citado na página 7.

MORAIS, E. C. Reconhecimento de padrões e redes neurais artificiais em predição de estruturas secundárias de proteínas. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação, 2010. Citado na página 12.

OLIVEIRA, F. A. D. de. Processamenti de linguagem natural: princípios básicos a implementação de um analisador sintático de sentenças da língua portuguesa. **Universidade Federal do Rio Grande do Sul**, 2011. Citado na página 5.

PAO, Y.-H. **Adaptive Pattern Recognition and Neural Networks**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0-201-12584-6. Citado na página 12.

PERNA, H. C. L. **Linguagens especializadas em corpora : modos de dizer e interfaces de pesquisa**. Edipucrs, 2010. ISBN 9788539700240. Disponível em: <https://books.google.com.br/books?id=2gV5PGfSk0QC>. Citado 2 vezes nas páginas 3 e 4.

REESE, R. M.; BHATIA, A. **Natural Language Processing with Java: Techniques for Building Machine Learning and Neural Network Models for NLP, 2Nd Edition**. 2nd. ed. [S.l.]: Packt Publishing, 2015. ISBN 1788993497, 9781788993494. Citado na página 4.

RICH, E. et al. **Inteligencia artificial**. McGraw-Hill, 1994. ISBN 9788448118587. Disponível em: <https://books.google.com.br/books?id=yunOPQAACAAJ>. Citado na página 5.

RUSSELL, S.; NORVIG, P. **Inteligência artificial**. [S.l.]: CAMPUS - RJ, 2013. ISBN 9788535211771. Citado 3 vezes nas páginas 3, 9 e 10.

SANTOS, M. C. de S. Análise e implementação de algoritmos de compressão de dados. **Centro Universitário Euripides de Marilia – UNIVEM**, 2018. Citado na página 10.

SETZER, V. W. **Alan Turing e a Ciência da Computação**. Depto. de Ciência da Computação da USP, 2006. Disponível em: <https://www.ime.usp.br/~vwsetzer/Turing-teatro.html>. Citado na página 3.

SILVA, J. R. C. da. **Processamento de Linguagem Natural (PLN)**. 2013. Disponível em: <http://jeiks.net/wp-content/uploads/2013/10/IntArt_PLN.pdf>. Acesso em: 23/12/2019. Citado na página 6.

SOUZA, F. G. P. de. Métodos universais de compressão de dados. **Instituto Brasileiro de Matemática, Estatística e Ciência da Computação, UNICAMP**, 1991. Citado na página 10.

SOUZA, O. d. et al. Um método de sumarização automática de textos através de dados estatísticos e processamento de linguagem natural. **Informação amp; Sociedade: Estudos**, v. 27, n. 3, dez. 2017. Disponível em: <https://periodicos.ufpb.br/ojs2/index.php/ies/article/view/32571>. Citado na página 8.

THANAKI, J. **Python Natural Language Processing: Advanced Machine Learning and Deep Learning Techniques for Natural Language Processing**. [S.l.]: Packt Publishing, 2017. ISBN 1787121429, 9781787121423. Citado 3 vezes nas páginas 4, 5 e 6.

TURBAN, E. et al. **Tecnologia da Informação para Gestão: Transformando os Negócios na Economia Digital**. [S.l.]: Bookman, 2010. ISBN 9788577805082. Citado na página 3.

UFSM, U. F. de S. M. **Compactação de Dados**. 2006. Disponível em: <http://coral.ufsm.br/tielletcab/Telframe/compacta.html>. Citado na página 10.

WALAA, M.; AHMED, H.; HODA, K. Sentiment analysis algorithms and applications: A survey. In: . [S.l.: s.n.], 2014. Citado na página 10.