

## Classificador Fuzzy-genético aplicado ao processamento de linguagem natural

Fernanda C. e Silva\* Rafael H. de Sousa\*  
Aleson G. S. Chaves\* Bruno H. G. Barbosa\*  
Danton D. Ferreira\*

\* *Departamento de Automática,  
Universidade Federal de Lavras,  
Lavras, MG, Brasil*

Emails: fernanda.silva10@estudante.ufla.br,  
rafael.sousa2@estudante.ufla.br, alesongsc@gmail.com,  
brunohb@ufla.br, danton@ufla.br

---

**Abstract:** Opinion mining analyzes opinions and feelings about an entity, which can be a product, a service, a person, etc. With the increasing use of the Internet, sentiment analysis has become an essential approach to analyzing the large amount of data generated. This analysis makes it possible to draw a profile of consumers, being a tool to assist companies in creating campaigns and improving their products. Several methods have been developed for the automatic classification of data in text format. The objective of this work is to design and evaluate a fuzzy classifier for mining opinion and classifying the general feeling of texts. For this, a database containing product reviews from the Epinions.com website was used. Before performing the classification, it was necessary to pre-process the data and extract characteristics, using two different methods. This work also proposes the use of the genetic algorithm to determine the characteristics that will be used by the fuzzy algorithm for classification, in order to maximize the accuracy value. The results obtained show a better accuracy for the classifier using characteristics extracted via Word2Vec when compared to the polarity method. In addition, the results obtained with the proposed method using 5 characteristics extracted via Word2Vec are superior to those obtained in other methods using 200 characteristics.

**Resumo:** A mineração de opinião analisa as opiniões e sentimentos sobre alguma entidade, podendo ser um produto, um serviço, uma pessoa, etc. Com o crescente uso da Internet, a análise de sentimentos tornou-se uma abordagem essencial para analisar a grande quantidade de dados gerados. Essa análise permite traçar um perfil dos consumidores, sendo uma ferramenta para auxílio das empresas na criação de campanhas e melhorias de seus produtos. Vários métodos foram desenvolvidos para a classificação automática de dados em formato de texto. O objetivo desse trabalho é projetar e avaliar um classificador *fuzzy* para mineração de opinião e classificação do sentimento geral de textos. Para isso, foi utilizada uma base de dados contendo revisões de produtos do site *Epinions.com*. Antes de realizar a classificação, foi necessário fazer o pré-processamento dos dados e a extração de características, com dois métodos diferentes. Esse trabalho também propõe a utilização do algoritmo genético para determinar as características que serão usados pelo algoritmo *fuzzy* para classificação, de forma a maximizar o valor da acurácia. Os resultados obtidos mostram uma melhor acurácia para o classificador usando características extraídas via *Word2Vec* quando comparado ao método por polaridades. Além disso, os resultados obtidos com o método proposto utilizando 5 características extraídas via *Word2Vec* é superior aos obtidos em outros métodos utilizando 200 características.

*Keywords:* Natural Language Processing; Fuzzy Logic; Genetic Algorithms; Feature Extraction  
*Palavras-chaves:* Processamento de Linguagem Natural; Lógica Fuzzy; Algoritmos Genéticos; Extração de Características.

---

## 1. INTRODUÇÃO

A área de Inteligência Artificial conhecida como Processamento de Linguagem Natural (PLN) estuda a capacidade e as limitações de uma máquina em entender a linguagem dos seres humanos (Zong and Hong (2018)). Com o crescente uso das mídias sociais na Internet, a mineração de opiniões se tornou uma abordagem essencial para analisar os dados, sendo utilizada em aplicações como precificação de produtos, previsão de mercado, previsão de eleições, inteligência competitiva, entre outras (Sun et al. (2016)).

Com a tendência das empresas disponibilizarem seus produtos para compra online, o aprendizado de máquina começou a ser aplicado para realizar o entendimento semântico de textos com o intuito de construir perfis de usuário e descobrir suas preferências, na expectativa de melhorar a recomendação de produtos e gerar uma melhor experiência de compra (Chen and Wang (2013)).

Muitos métodos de classificação foram desenvolvidos para a classificação automática de dados em formato de texto. As metodologias tradicionais de classificação como *Naive Bayes* (NB), *k-nearest neighbor* (KNN) e Máquina de vetores de suporte (SVM, do inglês Support Vector Machine) são utilizadas para realizar classificação de sentimentos em documentos, usando PLN (Li et al. (2018) e Vanaja and Belwal (2018)).

No artigo de Solangi et al. (2018) são revisadas as técnicas de PLN para mineração de opinião e análise de sentimentos. Já em seu trabalho, Tang et al. (2014) introduziram uma técnica de coleta de informações contextuais e de sentimentos das palavras, aprendendo a incorporação de palavras específicas de sentimentos. Eles aplicaram seu modelo para extrair sentimento de publicações realizadas no Twitter. Já Cui et al. (2006) trabalharam em análises de produtos online, classificando cerca de 100 mil revisões de produtos em duas classes principais: positiva e negativa.

No trabalho de Choudhary and Choudhary (2018) é feita a análise de opiniões sobre as mais recentes marcas de celulares. As opiniões foram coletadas diretamente do *Twitter*. Os resultados foram apresentados em gráficos e podem ser utilizados pelas marcas para melhorarem suas vendas. Já G. Chen and Xu (2016) observaram que um grande volume de vendas não gera necessariamente sentimentos positivos e vice-versa. A análise foi feita construindo-se perfis de consumidores de serviços de vendas *online*, de diferentes regiões. Esses trabalhos mostram como aplicações com PLN podem auxiliar empresas de comércio eletrônico.

No artigo de P. Pankaj and Soni (2019) é proposto um método geral para encontrar opinião contida em análises de produtos *online*, explorando a diferença nas estatísticas de duas compilações, um corpus específico de domínio e um corpus independente de domínio. Também discute técnicas e abordagens existentes para extração de características em análise de sentimentos e mineração de opinião.

No trabalho de Vanaja and Belwal (2018) foi feita a comparação entre a classificação realizada pelos algoritmos NB e SVM. Já Y. Liu and Shahbazzade (2018) compararam o algoritmo FL-SVM com os algoritmos NB, KNN e SVM

em diferentes conjuntos de dados e mostrou que o FL-SVM pode alcançar a melhor precisão de classificação de sentimentos, aumentando de 1% a 3% quando comparada aos outros algoritmos.

Um sistema de inferência neuro-fuzzy adaptável (ANFIS) é um tipo de rede neural artificial baseada no sistema de inferência *fuzzy* Takagi-Sugeno. A lógica *fuzzy*, quando comparada com a lógica booleana, permite mais representações do conhecimento em um ambiente de incerteza e imprecisão, modelando as interações e relacionamentos entre as variáveis do sistema. (Cordón et al. (2001)). A lógica booleana somente permite que as variáveis tenham 0 ou 1 como valores lógicos. Já a lógica *fuzzy* permite qualquer número real entre 0 e 1.

Os algoritmos genéticos (AG) são métodos de otimização global que funcionam com base na sobrevivência do mais apto e nos mecanismos da seleção natural (Padmaja and Hegde (2019)). Diversos trabalhos utilizam algoritmos genéticos para otimização dos resultados em diferentes tipos de problemas de otimização (Chen and Dai (2020)), classificação (Mortezanezhad and Daneshifar (2019) e Shi and Xu (2018)), entre outros.

Um Sistema Fuzzy Genético (SFG) é um sistema híbrido no qual um sistema *fuzzy* é otimizado por um processo de aprendizado genético. Uma otimização possível é aquela em que um algoritmo genético é usado para ajustar os diferentes componentes de um sistema baseado em regras *fuzzy* (Cordón et al. (2001)).

No trabalho de Padmaja and Hegde (2019) foi desenvolvido um SFG em que as regras *fuzzy* são geradas usando o método de classificação ANFIS. A saída do classificador ANFIS é fornecida como entrada para o AG encontrar os valores ideais para regras *fuzzy*.

De maneira semelhante, o trabalho apresentado neste artigo propõe um SFG, porém com uma abordagem diferente, utilizando o AG para determinar os valores que serão usados pelo ANFIS para classificação, de forma a maximizar o valor da acurácia. O trabalho está organizado da seguinte forma: a Seção II apresenta a pesquisa de vários artigos envolvendo análise de sentimentos em áreas relacionadas ao comércio eletrônico. A seção III apresenta as etapas para construção do SFG proposto. Já a análise quantitativa e comparativa do sistema proposto está detalhada na seção IV. Por fim, a conclusão do trabalho de pesquisa com sugestões para futuros trabalhos estão descritas na seção V.

## 2. MATERIAIS E MÉTODOS

### 2.1 Base de dados

A base de dados utilizada contém revisões de produtos do site *Epinions.com*, que foi um serviço online criado em 1999, e oferecia avaliações de produtos feitas pelos próprios consumidores. O site foi desativado em 2014 e substituído pelo site *Shopping.com*. Apesar de as avaliações serem antigas, a estrutura textual é semelhante aos comentários encontrados hoje em diversos sites de compras *online*.

A base passou por um processo de balanceamento, para que contivesse o mesmo número de avaliações positivas e negativas. Foram selecionadas 691 postagens que recomendam automóveis Ford e 691 postagens que não recomendam automóveis Ford. No total, foram analisadas 1382 amostras. As avaliações analisadas estão em formato textual, em inglês. Os dados podem ser divididos em duas classes: Pos (críticas que expressam sentimentos positivos ou favoráveis) e Neg (críticas que expressam sentimentos negativos ou desfavoráveis).

## 2.2 Pré-processamento

Os dados foram extraídos de um arquivo, em que cada comentário estava em uma linha do arquivo. Considerando que textos são dados não estruturados e podem conter caracteres especiais e sinais de pontuação, é importante realizar o pré-processamento dos dados antes do início do processo de classificação. Os caracteres indesejados, tais como #, @ e / foram removidos.

## 2.3 Tokenização e transformação dos dados via extração de polaridades

A etapa de tokenização é utilizada para dividir cada amostra em *tokens* para o processo de análise e mineração de texto. Inicialmente, as amostras são segmentadas e são localizados os limites para formar os *tokens*. O limite foi determinado pelo início e fim de uma palavra. Para realizar esse processo, utilizou-se a biblioteca *Textblob*, que é escrita em *Python* e faz processamento de dados em formato textual.

Como nem toda palavra da frase expressa uma opinião, a biblioteca também foi usada para realizar a marcação gramatical de cada *token*, associando a classe gramatical como pronome, verbo, adjetivo, advérbio. Essa classificação é conhecida como *part-of-speech tag*.

Após essa etapa foram selecionados os adjetivos e advérbios, por serem as palavras relevantes para a análise. Para determinar a polaridade de cada comentário, foi analisada cada palavra restante, usando o dicionário *SentiWordNet 3.0*.

Para análise, cada palavra foi reduzida ao seu radical, removendo prefixos, sufixos e outros modificadores da palavra para aumentar a chance dessa palavra ser encontrada no dicionário. Para cada palavra são buscados seus *synsets*, que são os sinônimos da palavra analisada. O *synset* possui três valores, o de positividade, o de negatividade e o de objetividade.

Para cada palavra, é realizada a soma da parte positiva menos a soma da parte negativa de cada *synset*. Depois esse valor é dividido pela quantidade de *synsets* da palavra. O resultado dessas etapas é um vetor contendo os adjetivos e advérbios do comentário original e, para cada um, a classificação gramatical e o valor de polaridade calculado.

Os valores obtidos foram então combinados para gerar as 20 características listadas a seguir:

- (1) Soma dos adjetivos positivos;
- (2) Soma dos adjetivos positivos dividido pelo número de palavras extraídas do comentário original para classificação;

- (3) Quantidade de adjetivos positivos extraídos do comentário;
- (4) Soma dos adjetivos negativos;
- (5) Soma dos adjetivos negativos dividido pelo número de palavras extraídas do comentário original para classificação;
- (6) Quantidade de adjetivos negativos extraídos do comentário;
- (7) Soma dos advérbios positivos;
- (8) Soma dos advérbios positivos dividido pelo número de palavras extraídas do comentário original para classificação;
- (9) Quantidade de advérbios positivos extraídos do comentário;
- (10) Soma dos advérbios negativos;
- (11) Soma dos advérbios negativos dividido pelo número de palavras extraídas do comentário original para classificação;
- (12) Quantidade de advérbios negativos extraídos do comentário;
- (13) Polaridade resultante para adjetivos;
- (14) Polaridade resultante para advérbios;
- (15) Quantidade de adjetivos positivos menos a quantidade de adjetivos negativos;
- (16) Quantidade de advérbios positivos menos a quantidade de advérbios negativos;
- (17) Valor do adjetivo mais positivo ou do mais negativo;
- (18) Valor do advérbio mais positivo ou do mais negativo;
- (19) Quantidade de palavras extraídas do comentário original;
- (20) Quantidade de palavras do comentário original.

O processo é repetido para todos os comentários e a matriz de características obtida é salva em um documento *txt*, que é modificado posteriormente, para inserção da 21ª coluna, com o valor da polaridade de cada comentário.

## 2.4 Tokenização e transformação dos dados via *Word2Vec*

O *Word2Vec* é empregado, em geral, para *word embedding*. Ele é capaz de extrair conhecimento semântico dos textos, além de realizar a análise sintática e morfológica. O *Word2Vec* é eficiente em termos computacionais devido ao seu modelo ser redes neurais com poucas camadas, sendo eficientes para treinamentos em grandes conjuntos de dados textuais. Apresenta, a capacidade de ser treinado rapidamente e produz resultados com uma boa acurácia.

O *Word2Vec* tem a característica de mapear palavras que apresentam similaridade em posições próximas dos vetores, ou seja, possuem valores próximos de caracterização (Mikolov et al. (2016)).

Para a implementação do *Word2Vec* foi utilizada a linguagem *Python* e suas bibliotecas *open source* para aprendizagem de máquina como *pandas*, *numpy*, *matplotlib*, *seaborn*, *string* e *nltk*. Essas bibliotecas são comumente utilizadas para o desenvolvimento de projetos na área de inteligência artificial, tal como em aplicações de processamento de linguagem natural.

Após a etapa de extração de caracteres via *Word2Vec* seguiram-se as etapas de pré processamento, tokenização e *stemming*, de forma similar à extração de características por polaridade. Porém o *Word2Vec* utilizou as bibliote-

cas *NLTK* e *numpy* para realização destas operações. Já para a extração de características foi utilizada a biblioteca *gensim*. Esta implementação do *Word2Vec* permite a configuração do modelo em diversos parâmetros.

A arquitetura do *Word2Vec* utilizada foi do tipo *skip-gram*, que faz a previsão das palavras de contexto dada uma palavra de origem. Neste caso, a rede neural vai ter como entrada o vetor com uma palavra, e a saída será os vetores com as palavras de contexto. Após ser treinada (no caso deste trabalho por 20 gerações), apresenta como resultado a transformação das palavras tokenizadas em valores numéricos, formando um vetor com as características das palavras normalizadas entre -1 e 1.

O *Word2Vec* gera um vetor com  $n$  características para cada comentário analisado, sendo esse valor escolhido pelo usuário. Esta é a representação vetorial das palavras das sentenças, sendo os valores numéricos (características) gerados seguindo as análises sintáticas e morfológicas de acordo com aspectos construtivos do modelo. Este apresenta a capacidade de realizar a composicionalidade. Ao adicionar-se dois vetores de palavras resulta em um vetor que é uma composição semântica de palavras individuais, por exemplo, "homem" + "real" = "rei" (Mikolov et al. (2013) e Mikolov et al. (2016)).

A análise de sentimentos é realizada pelos classificadores que fazem a leitura dos valores gerados pelo *Word2Vec*.

### 2.5 Algoritmo Genético

Algoritmos genéticos são técnicas de otimização desenvolvidas baseadas na teoria da evolução de Darwin, mais especificamente, no processo de seleção natural, onde os indivíduos com maior aptidão ao ambiente sobrevivem e se sobressaem sobre os menos aptos que são eliminados (Silveira and Barone (1998)).

Estes algoritmos de acordo com Choi et al. (2016) seguem uma sequência de processos de evolução dos indivíduos contidos na população durante iterações denominadas gerações. Durante cada geração os indivíduos passam por processos de *crossover* e mutação, recebem valores de aptidão de acordo com uma função de *fitness* e são selecionados com base nestes valores para compor a população na próxima geração.

O algoritmo genético construído para esse trabalho recebe como entradas a quantidade de características, a função *fitness*, o tamanho da população, a quantidade de indivíduos que serão candidatos a pais, a taxa de mutação, a quantidade de gerações e o valor de erro mínimo.

Antes de iniciar a execução do processo genético, o algoritmo importa os dados, os valores são normalizados e as amostras são embaralhadas, para que haja amostras positivas e negativas nos dados de treino, validação e teste. Os dados são divididos segundo a proporção: 60% treino, 20% validação e 20% teste. Então a população inicial é gerada selecionando para cada indivíduo as características que vão formar seu genótipo de forma aleatória, sendo cada um desses indivíduos composto de 5 valores inteiros.

Para cada geração é calculado o valor do *fitness* de acordo com a função passada e as amostras. Nesse caso, a função *fitness* é o próprio ANFIS. Depois, a população é ordenada

de acordo com o valor calculado do *fitness*, em ordem decrescente, já que o objetivo é maximizar a acurácia.

O algoritmo genético calcula a probabilidade de sorteio de cada pai, baseado no valor do ranking dos indivíduos, utilizando-se o método da roleta viciada. Para determinar os indivíduos da próxima geração, é utilizado elitismo, ou seja, o melhor indivíduo sempre permanece na população.

Todos os indivíduos candidatos a pais também permanecem na população e os filhos substituem os indivíduos que não foram selecionados como candidatos à pais. Cada par (pai, mãe) selecionado gera dois filhos através do *crossover* de um ponto, sendo que o ponto de corte é escolhido de forma aleatória. O *crossover* se repete até que seja obtido o número de filhos predeterminado. A mutação é feita gene a gene, substituindo-se o valor do gene a ser mutado por um número aleatório dentro do intervalo válido correspondente as características (inteiros de 1 a 20). Não ocorre mutação no melhor indivíduo, o que garante que a próxima geração terá um desempenho no mínimo, igual a população anterior. Os parâmetros utilizados são mostrados na Tabela 1.

Tabela 1. Tabela de Parâmetros aplicados ao Algoritmo Genético.

Parâmetros	Valores
Geração da População Inicial	Aleatória
Número de Características	5
Tipo de Característica	Números Inteiros
Número de Indivíduos	12
Número de Filhos Gerados	6
Taxa de Mutação	40%
Número de Gerações	12
Método de Seleção	Roleta Viciada
Método de Crossover	Corte em um ponto aleatório
Método de Mutação	Gene a gene
Número de Indivíduos Elite	1

### 2.6 Classificador

O modelo utilizado nesse trabalho é aquele em que, a partir de amostras de textos, o classificador realiza a classificação das amostras em duas categorias: negativo ou positivo. O classificador proposto é *fuzzy* genético.

O classificador, de acordo com o algoritmo, transforma os valores recebidos dos vetores de características *word embedding* nos valores 0 ou 1. Nesta implementação, o valor 0 é atribuído para os sentimentos positivos e o valor 1 é atribuído para os sentimentos negativos. O classificador faz então a comparação com a classificação original da base de dados e analisa estatisticamente os acertos.

Para o SFG foram utilizadas 20 características e, após a seleção do AG, 5 características foram enviadas para classificação pelo ANFIS, cuja configuração pode ser vista na tabela 2. Além do SFG, para fins de comparação e análise de desempenho, foram utilizados para a classificação, os algoritmos de Regressão Logística, *Support Vector Machine* e *Random Forest*, que foram implementados utilizando a biblioteca *python sklearn*. Estes classificadores utilizaram 200 características geradas pelo *Word2Vec* e não foi feita a seleção das características utilizadas.

O algoritmo *Random Forest*, cria uma floresta de árvores de decisão aleatoriamente. Tal combinação de árvores de

decisão é um dos mais simples algoritmos de aprendizagem. A classificação é feita ordenando os dados, criando uma árvore de decisão da raiz para as folhas. O algoritmo *Random Forest* busca a melhor característica em um subconjunto aleatório das características (Lan and Pan (2019)).

O algoritmo de Regressão Logística fornece um resultado binomial e a probabilidade do evento ocorrer ou não, ou seja, retorna como resultado 0 ou 1. A regressão logística apresenta como vantagens, a simplicidade de implementação, eficiência computacional, a rapidez no treinamento e a facilidade de regularização. Ele é capaz de resolver problemas de escala industrial (Ray (2019)).

O algoritmo *Support Vector Machine* pode ser utilizado tanto para regressão quanto para classificação. Este é um algoritmo de aprendizado supervisionado que faz a separação de um conjunto de objetos de diferentes classes através de um plano de decisão. Quando os conjuntos de objetos não são linearmente separáveis, utiliza-se funções matemáticas complexas chamadas *kernels* para separá-los (Silva et al. (2015) e Ray (2019)).

## 2.7 ANFIS

O ANFIS é utilizado como função *fitness* do algoritmo genético (AG). As entradas que o ANFIS recebe do AG são as características definidas pelo indivíduo em avaliação. De acordo com o tamanho da população, é gerado um ANFIS para cada indivíduo com número de variáveis correspondente ao número de características não repetidas contidas no indivíduo, e então, é feita a avaliação de todos os indivíduos da população.

Os dados são divididos segundo a proporção: 60% treino, 20% validação e 20% teste. O sistema de inferência fuzzy (*fis*, do inglês fuzzy inference system) inicial é gerado via *grid partition*, com função de pertinência *gbell*. Para cada entrada foram consideradas duas funções de pertinência. Todos os treinamentos foram feitos com 200 épocas e com objetivo de minimizar a raiz do erro quadrado médio (*RMSE*), considerando como critério de parada antecipada *RMSE* igual a zero. Foi usado o método de otimização *hybrid*, que utiliza mínimos quadrados e gradiente descendente para atualização de parâmetros consecuentes e antecedentes.

Depois de terminar o treino do ANFIS, todo o banco de dados é avaliado considerando os conjuntos de treino, validação e teste. Para cada amostra é feita a comparação entre a previsão de polaridade do ANFIS com o valor de polaridade real. O valor da acurácia é calculado de acordo com a equação 1, e corresponde ao valor de *fitness* de cada indivíduo. Os parâmetros utilizados no ANFIS são mostrados na Tabela 2.

## 3. ANÁLISE E DISCUSSÃO DOS RESULTADOS

O algoritmo genético-fuzzy foi usado para determinar as características mais apropriadas e realizar a classificação de sentimentos dos comentários do banco de dados *Epinions*, utilizando os métodos de extração de características *Word2vec* e Extração por polaridades.

Tabela 2. Tabela de Parâmetros aplicados ao Anfis.

Parâmetros	Valores
Número Máximo de Entradas	5
Percentual de Dados para Treino	60%
Percentual de Dados para Verificação	20%
Percentual de Dados para Validação	20%
Tipo de Fis utilizado	Grid Partition
Número de Funções de Pertinência por Entrada	2
Tipo de Função de Pertinência	Gbellmf
Número de Épocas de Treinamento	200
Método de Otimização	Hybrid

Para análise dos resultados foram utilizados os valores da acurácia (Equação 1), precisão (Equação 2), recall (Equação 3) e F1 (Equação 4), onde TP, FP, TN, FN representam as instâncias Verdadeiras Positivas, Falso Positivas, Verdadeiras Negativas e Falso Negativas, respectivamente. O valor F1 é obtido usando os valores de precisão e recall, gerando um número único que indica a qualidade geral do modelo. As fórmulas de cada equação anteriormente descritas são:

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precisão = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = \frac{2 * Precisão * Recall}{Precisão + Recall} \quad (4)$$

Os resultados obtidos são mostrados nas subseções a seguir.

### 3.1 Resultados utilizando a análise de polaridade das palavras como método de extração de características

O gráfico mostrado na Figura 1 representa o desempenho do melhor indivíduo de cada geração do algoritmo de acordo com a acurácia (equação 1). O gráfico da Figura 2 mostra quais características foram usadas no classificador em cada geração pelo melhor indivíduo, onde cada linha do gráfico corresponde a um gene.

Observando os gráficos é possível perceber que as alterações nas características empregadas geram alteração na acurácia do algoritmo. O resultado na primeira geração já é otimizado, uma vez que todos os indivíduos da população foram testados e o resultado mostrado corresponde ao indivíduo com maior acurácia dentre eles, e conseqüentemente, ao longo das outras gerações foram obtidos resultados ainda melhores, gerando um incremento adicional na acurácia de 2%. É importante citar que usar o ANFIS para as 20 características é inviável devido ao tempo de processamento e custo computacional.

O conjunto de características selecionadas, os valores da acurácia e F1 obtidos são mostrados na Tabela 3.

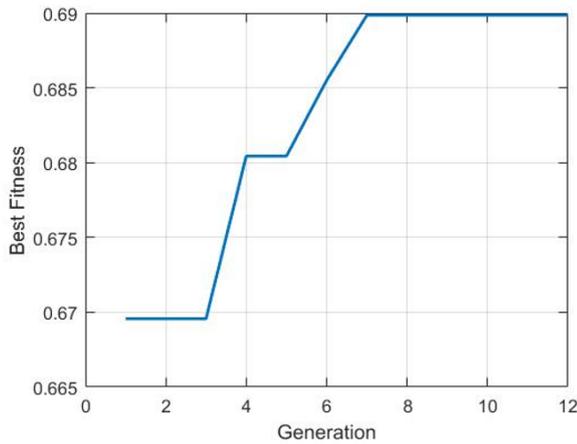


Figura 1. Evolução da acurácia da melhor solução - Polaridade de palavras.

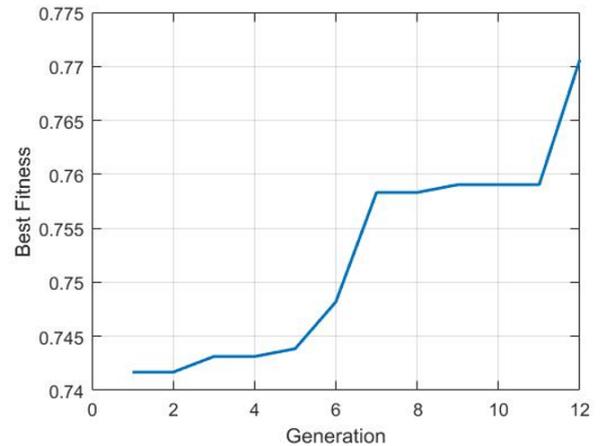


Figura 3. Evolução da acurácia da melhor solução - Word2vec.

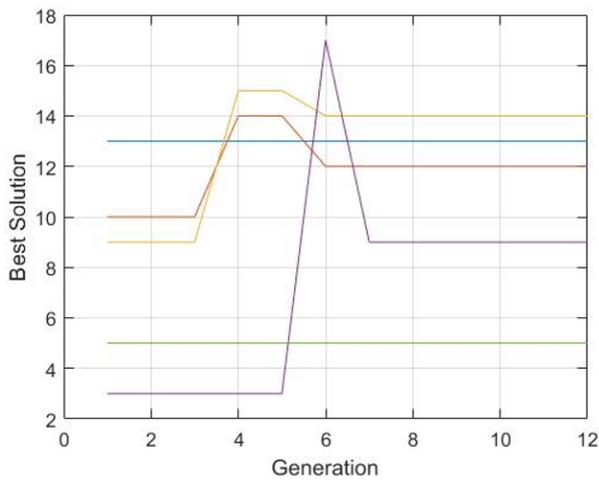


Figura 2. Evolução das características selecionadas - Polaridade de palavras.

### 3.2 Resultados utilizando o Word2Vec como método para extração de características

O gráfico mostrado na Figura 3 representa o desempenho do melhor indivíduo de cada geração do algoritmo. O gráfico da Figura 4 mostra quais características foram usadas no classificador em cada geração, onde cada linha representa um gene do indivíduo.

De maneira semelhante ao resultado exposto anteriormente para o método de extração de polaridades, é possível perceber que as alterações nas características empregadas geram alteração na acurácia do algoritmo. Entre a primeira geração e a última houve um incremento na acurácia de 3% (Além da otimização inicial da primeira geração). O conjunto de características selecionadas, os valores da acurácia e  $F1$  obtidos são mostrados na Tabela 4.

Tabela 3. Valores obtidos com Polaridade de palavras.

Geração	Características Selecionadas	Acurácia	F1
Primeira	3, 5, 9, 10, 13	0,6696	0,6864
Última	5, 9, 12, 13, 14	0,6899	0,7040

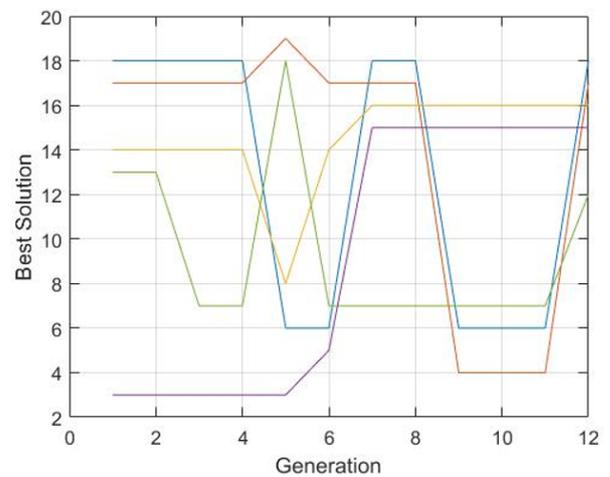


Figura 4. Evolução das características selecionadas - Word2vec.

É notável a ocorrência de maiores variações do conjunto de características que melhoram a acurácia do classificador para o método de extração usando o *Word2Vec* do que para o método de extração de polaridades (Figuras 2 e 4). Isto ocorre, principalmente, devido ao fato de as características geradas pelo método de extração de polaridades possuírem uma maior correlação entre si do que as características geradas pelo *Word2Vec*.

Nota-se ainda uma melhor acurácia, em geral, para o classificador usando características extraídas via *Word2Vec* do que para o modelo usando características extraídas via o método de polaridades, o que pode ser observado comparando as Figuras 1 e 3.

A partir dos valores de acurácia e  $F1$  obtidos para o algoritmo proposto com AGs e ANFIS, é possível fazer a comparação com diferentes algoritmos aplicados na análise

Tabela 4. Valores obtidos com Word2vec.

Geração	Características Selecionadas	Acurácia	F1
Primeira	3, 13, 14, 17, 18	0,7417	0,7411
Última	12, 15, 16, 17, 18	0,7706	0,7766

de sentimentos. A Tabela 5 mostra os dados de  $F1$  obtidos utilizando o método da extração de polaridades (20 características geradas) e o método de extração via *Word2Vec* (20 características geradas) para o SFG proposto, além de outros métodos de análise de sentimentos automáticos como a SVM, *Random Forest* e Regressão Logística, todos aplicados a base *Epinions 3* e com 200 características extraídas via *Word2Vec*.

Utilizou-se o *Word2Vec* para extrair 200 características a serem utilizadas pelos métodos de análise automáticos, uma vez que foi notada uma grande queda de desempenho destes algoritmos quando usando apenas 20 características como em nosso SFG.

Tabela 5. Comparação com outros métodos.

Algoritmos - Métodos de Extração de Características	F1
SFG – Extração de Polaridades	0,7040
SFG – Word2Vec	0,7766
Regressão Logística – Word2Vec	0,7491
Support Vector Machine – Word2Vec	0,7606
Random Forest – Word2Vec	0,7563

Pode-se verificar que o SFG proposto neste trabalho, que utilizou como banco de dados características extraídas via método de extração de polaridades, possui o pior desempenho, com o valor  $F1$  de aproximadamente 0,70. É possível que isso tenha ocorrido devido ao fato de as características obtidas possuírem, em geral, uma alta correlação, já que muitas delas são derivadas de combinações entre si. É possível ainda que um melhor conjunto de características possa ser selecionado, uma vez que foram utilizadas poucas gerações na busca da melhor solução pelo SFG, devido ao tempo que o algoritmo demanda.

Para o banco de dados com características extraídas via *Word2Vec*, o SFG conseguiu determinar um conjunto de características que apresenta melhor desempenho com relação a todos os algoritmos automáticos de análise de sentimentos presentes na Tabela 5, o que revela o potencial aplicável do SFG proposto. Além disto, o SFG nesta situação também utilizou um número pequeno de gerações, o que permite dizer que, possivelmente, poderiam ocorrer resultados ainda melhores na acurácia.

Por fim, o SFG permite a redução da complexidade do problema, uma vez que atingiu tais resultados utilizando apenas 5 das características extraídas, enquanto os métodos de análise de sentimentos automáticos utilizados na comparação utilizaram 200 características para atingir os resultados mostrados na Tabela 5.

#### 4. CONCLUSÃO

Atualmente a análise de sentimentos é uma área que tem despertado bastante interesse por suas diversas aplicações. Neste trabalho é proposto um classificador *fuzzy* para mineração de opinião e classificação do sentimento geral de textos. Os resultados obtidos mostram uma melhor acurácia para o classificador usando características extraídas via *Word2Vec* quando comparado ao método por polaridades. O  $F1$  Score obtido com o SFG utilizando 5 características extraídas via *Word2Vec* é superior aqueles obtidos pelos métodos NB, KNN e SVM utilizando 200 características.

Para trabalhos futuros, é possível estudar maneiras de selecionar um melhor conjunto de características, aumentar o número de gerações na busca da melhor solução pelo SFG ou reduzir o tempo de processamento, pois foram fatores limitantes para o desenvolvimento deste trabalho. Além disso, o classificador pode ser adaptado para atribuir diferentes classes às avaliações, considerando os graus de subjetividade existentes, por exemplo, se um produto é bom ou muito bom, criando uma espécie de *ranking* das opiniões.

#### AGRADECIMENTOS

Os autores agradecem à CAPES e FAPEMIG por apoiarem este trabalho.

#### REFERÊNCIAS

- Chen, L. and Wang, F. (2013). Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowledge-Based Systems*, 50, 44–59.
- Chen, X. and Dai, Y. (2020). Research on an improved ant colony algorithm fusion with genetic algorithm for route planning. *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 1, 1273–1278.
- Choi, K., Jang, D., Kang, S., Lee, J., Chung, T., and Kim, H. (2016). Hybrid algorithm combining genetic algorithm with evolution strategy for antenna design. *IEEE Transactions on Magnetics*, 52(3), 1–4.
- Choudhary, M. and Choudhary, P.K. (2018). Sentiment analysis of text re-viewing algorithm using data mining. *2018 International Conference Smart Systems and Inventive Technology (ICSSIT)*, 532–538.
- Cordón, O., Herrera, F., Hofmann, F., and Magdalena, L. (2001). Genetic fuzzysystems. evolutionary tuning and learning of fuzzy knowledge bases. *World Scientific*.
- Cui, H., Mittal, V., and Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2, AAAI'06*, 1265–1270. AAAI Press.
- G. Chen, Y.W. and Xu, X. (2016). An analysis of the sales and consumer preferences of e-cigarettes based on text mining of online reviews. *2016 3rd International Conference on Systems and Informatics (ICSAI)*, 1045–1049.
- Lan, H. and Pan, Y. (2019). A crowdsourcing quality prediction model based on random forests. *Proceedings - 18th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2019*, 315–319.
- Li, M.Y., Kok, S., and Tan, L. (2018). Don't classify, translate: Multi-level e-commerce product categorization via machine translation. *CoRR*, abs/1812.05774.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2016). Distributed representations of words and phrases and their compositionality. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1389–1399.

- Mortezanezhad, A. and Daneshifar, E. (2019). Big-data clustering with genetic algorithm. *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 702–706.
- P. Pankaj, P. Pandey, M. and Soni, N. (2019). Sentiment analysis on customer feedback data: Amazon product reviews. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 320–322.
- Padmaja, K. and Hegde, N.P. (2019). Twitter sentiment analysis using adaptiveneuro-fuzzy inference system with genetic algorithm. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 498–503.
- Ray, S. (2019). A quick review of machine learning algorithms- proceedings of the international conference on machine learning. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects, COM-IT-Con 2019*, 35–39.
- Shi, H. and Xu, M. (2018). A data classification method using genetic algorithm and k-means algorithm with optimizing initial cluster center. *2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET)*, 224–228.
- Silva, R.D., Thé, G.A., and De Medeiros, F.N. (2015). Geometrical and statistical feature extraction of images for rotation invariant classification systems based on industrial devices. *21st International Conference on Automation and Computing: Automation, Computing and Manufacturing for New Economic Growth, ICAC 2015*, 1–6.
- Silveira, S.R. and Barone, D.A.C. (1998). Jogos educativos computadorizados utilizando a abordagem de algoritmos genéticos. *Universidade Federal do Rio Grande do Sul. . . .*
- Solangi, Y.A., Solangi, Z.A., Aarain, S., Abro, A., Mallah, G.A., and Shah, A. (2018). Review on natural language processing (nlp) and its toolkits for opinion mining and sentiment analysis. In *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 1–4.
- Sun, S., Luo, C., and Chen, J. (2016). A review of natural language processing techniques for opinion mining systems. *Information Fusion*, 36.
- Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 1, 1555–1565.
- Vanaja, S. and Belwal, M. (2018). Aspect-level sentiment analysis on e-commerce data. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1275–1279.
- Y. Liu, J.L. and Shahbazzade, S. (2018). Sentiment classification of e-commerce product quality reviews by flsvm approaches. *2018 IEEE 17th International Conference on Cognitive Informatics Cognitive Computing (ICCI\*CC)*, 292–298.
- Zong, Z. and Hong, C. (2018). On application of natural language processing in machine translation. In *2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, 506–510.