PROJETO E DESENVOLVIMENTO DE UM SISTEMA PARA DEFINIÇÃO DE ASPECTOS E ANÁLISE DE SENTIMENTOS EM TEXTOS

Evandro Franzen[1], Frederico Jacobi Sausen[2]

Resumo: Ambientes virtuais, como redes sociais e *blogs*, apresentam textos e comentários que geralmente contêm opiniões. A partir desses ambientes, é possível aplicar a mineração de opiniões ou análise de sentimento com o objetivo de identificar automaticamente a opinião nas expressões com flexibilidade e rapidez, realizando avaliações, polarizando sentimentos em uma grande massa de conteúdo na forma de texto. Além da classificação automática dos textos, é necessário também definir o alvo ou aspecto relacionado ao comentário, o que por vezes se torna uma tarefa manual e cansativa. Este artigo apresenta uma ferramenta que permite a criação de ontologia e associação dos termos aos textos coletados na *web*. O sistema permite também a aplicação de algoritmos para avaliar a viabilidade de efetuar a classificação automática das publicações analisadas. São descritos os recursos existentes no *software* e os resultados de testes realizados com o objetivo de validar a ferramenta.

Palavra-chave: Mineração de opiniões. Classificação da polaridade. Aprendizado de máquinas. Algoritmo de classificação.

1 INTRODUÇÃO

Com o crescimento das mídias sociais, *blogs* e *web*, a relação entre usuários está cada vez mais próxima e rápida. Devido a essa proximidade, as opiniões de alguém podem se disseminar de forma rápida, tanto as boas quanto as más, podendo influenciar a organização nos processos de tomada de decisão. O gasto das organizações em *marketing* nas redes sociais, *web*, *e-mail* e outros meios é cuidadosamente examinado, pois a avaliação de apenas um usuário pode se tornar catastrófica para a reputação do produto ou da empresa.

Uma opinião é definida por dois elementos em um documento: o alvo da opinião e o sentimento expresso sobre o alvo. O documento pode ser qualquer texto em linguagem natural e o alvo também é conhecido como entidade, tópico ou aspecto sobre o qual foi manifestada a opinião de um produto, serviço, entre outros, sendo polarizado entre bom e ruim (CARVALHO, 2014).

A Mineração de Dados (MD) tem como objetivo principal transformar os dados armazenados em conhecimento, expresso em termos de formalismos de representação, tal como regras e relações entre dados. Uma das formas de tornar os dados armazenados em conhecimento é utilizando algoritmos, de diversas formas, organizando processos, etapas de testes e treinamento até chegarmos ao resultado final (REZENDE et al., 2003).

---

1 Mestre em Computação pela Universidade Federal do Rio Grande do Sul (2002). Professor assistente do Centro Universitário UNIVATES.

2 Acadêmico do Centro Universitário UNIVATES.

Este artigo documenta o desenvolvimento de um sistema para identificar aspectos em textos e para aplicar algoritmos de aprendizado de máquina. São utilizadas técnicas de mineração de opiniões, em que os dados são dispostos para treino em um *software* elaborado pelo autor principal do artigo, no qual são atribuídas as polaridades dos textos manualmente criando um modelo de treino, modelo que poderá ser importado na ferramenta de mineração de dados Weka[3] como base para atribuição das polaridades.

O trabalho tem como objetivo apresentar os principais conceitos e recursos relacionados aos recursos e módulos da ferramenta. Também visa a desenvolver e testar uma ferramenta que permita a definição de aspectos, por exemplo, marca, modelo e tipo de produto, e classes, como positivo e negativo, de forma prática no texto manualmente. O sistema foi integrado a outra ferramenta de mineração de dados para o pré-processamento, extração e pós-processamento dos dados. A interface da ferramenta possibilita a visualização dos aspectos e das classes do conjunto de dados analisados, por meio de avaliações e resultados das classificações.

Decidir entre qual algoritmo utilizar na mineração de dados é uma tarefa importante e que requer testes para identificar qual técnica se adapta melhor ao problema que é preciso resolver. Para o uso correto dos algoritmos classificadores, é necessário ter um conjunto de dados-treino rotulados, sendo um processo manual e exaustivo ter que atribuir aspectos e classes ao volume de dados suficientes para que se tenha algum resultado significativo.

Construir uma ferramenta mais amigável ao usuário para classificação do conjunto de dados extraídos da *web* facilita o processo como um todo da mineração de dados. A ferramenta desenvolvida, denominada Análise em Mineração de Dados (AMD), permite a definição de estruturas variáveis para cada cenário a ser pesquisado, tornando inteligente a ferramenta para modelos de treinos, podendo não só pesquisar produtos eletrônicos, mas também opiniões de assuntos da atualidade, por exemplo: copa, eleições, entre outros.

A seção 2 deste artigo descreve os pressupostos teóricos do trabalho, incluindo os fundamentos sobre a descoberta de conhecimento e mineração de dados. A seguir, são apresentadas as principais tecnologias e a metodologia adotada no trabalho e a seção 4 apresenta os resultados, a implementação da ferramenta e os testes realizados.

2 DESCOBERTA DE CONHECIMENTO

Ainda existem dificuldades na descoberta de conhecimento em banco de dados, baseada nas grandes informações em banco de dados que as empresas possuem, relacionadas a fatores como: mineração de dados em que há falta de conhecimento da existência de técnicas, técnicas complexas, falta de ferramentas adequadas, alto custo das ferramentas de mineração ou escolha errada da técnica conforme problema a ser resolvido.

A mineração de dados está em uma das suas fases do processo *Knowledge Discovery in Databases* (KDD), na qual ocorre a aplicação de algoritmos que identificam padrões válidos, novos, potencialmente úteis e compreensíveis. O processo se preocupa com o desenvolvimento de métodos e técnicas para fazer sentido aos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

O método tradicional de transformar os dados em conhecimento baseia-se na análise manual e interpretativa. Porém, a descoberta de conhecimento refere-se ao processo global de descoberta de conhecimento a partir de dados e a mineração de dados refere-se a uma etapa nesse processo. A mineração de dados é uma aplicação de algoritmos específicos para extração de padrões de dados,

3 Disponível em: <http://www.cs.waikato.ac.nz/ml/weka>.

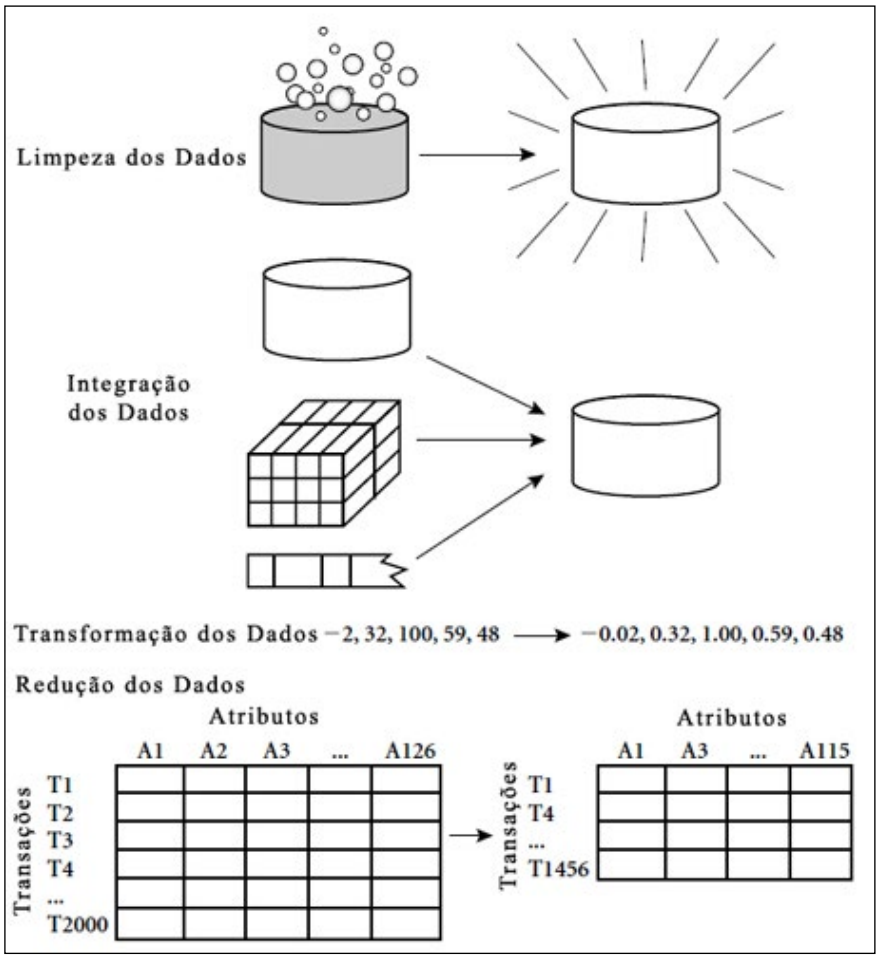
existindo distinção entre o processo KDD e a etapa de mineração de dados. (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

No processo de mineração de dados, a coleta, a aplicação dos algoritmos e a visualização dos dados são divididas em diversas etapas, não sendo um sistema automático. Nessas etapas são feitos o conhecimento do domínio do que será minerado, o pré-processamento, a extração de padrões, o pós-processamento e a utilização do conhecimento.

Conhecimento no conteúdo, domínio abordado, definição das metas e objetivos são traçados nessa etapa, tendo papel importante no fornecimento de conhecimento do conteúdo. Antes de aplicar a mineração de dados, precisamos saber com que tipo de dados se está trabalhando. Além de definir as informações corretas, as definições das informações incorretas também precisam de atenção, assim como os nulos, em branco e *emotions* devem ser categorizados.

Conforme a Figura 1, o pré-processamento consiste na limpeza dos dados. Diversas vezes os dados são encontrados com inconsistências, ocorrendo então a avaliação de dados, evitando futuros problemas nos algoritmos. Na limpeza, os dados podem ser removidos, agrupados e até ganhar atributos (CAMILO, 2009).

Figura 1 - Atividade do pré-processamento



Fonte: Camilo (2009).

Nos textos existem termos frequentes que não carregam nenhuma informação de maior relevância, as *stopwords*, as quais são compostas por palavras de diversas classes gramaticais, como: artigos, preposições, conjunções, pronomes e advérbios. A remoção das *stopwords* objetiva eliminar termos não representativos ao texto. Essa técnica também é considerada como compressão de textos, já que reduz as palavras analisadas no texto e o número de palavras a serem armazenadas no banco de dados (DIAS; DE GOMENSORO MALHEIROS, 2005).

Na etapa de extração de padrões, os objetivos traçados na identificação do problema tendem a ser cumpridos. A escolha do algoritmo está baseada nos resultados a serem encontrados. Algoritmos mais simples são de melhor interpretação, já algoritmos complexos são utilizados por pesquisadores.

No pós-processamento, a busca por conhecimento no banco de dados é exaustiva. O resultado obtido pelo algoritmo pode às vezes conter inúmeros padrões, com redundâncias, muitos desses sem valor algum ao usuário. Este quer encontrar uma pequena listagem, algo que ele consiga entender e seja objetivo. Então, disponibilizar grande número de padrões pode ser decepcionante à sua expectativa. Nesse processo, as informações devem estar organizadas conforme objetivo inicial da mineração.

No cenário das redes sociais, a mineração de opinião automática começa a criar forma. Começam a surgir mecanismos de obtenção e manipulação desse grande volume de informações, algoritmos que extraem os dados, agrupam as informações e disponibilizam de forma prática, em tempo muito menor, com número de conhecimento muito maior.

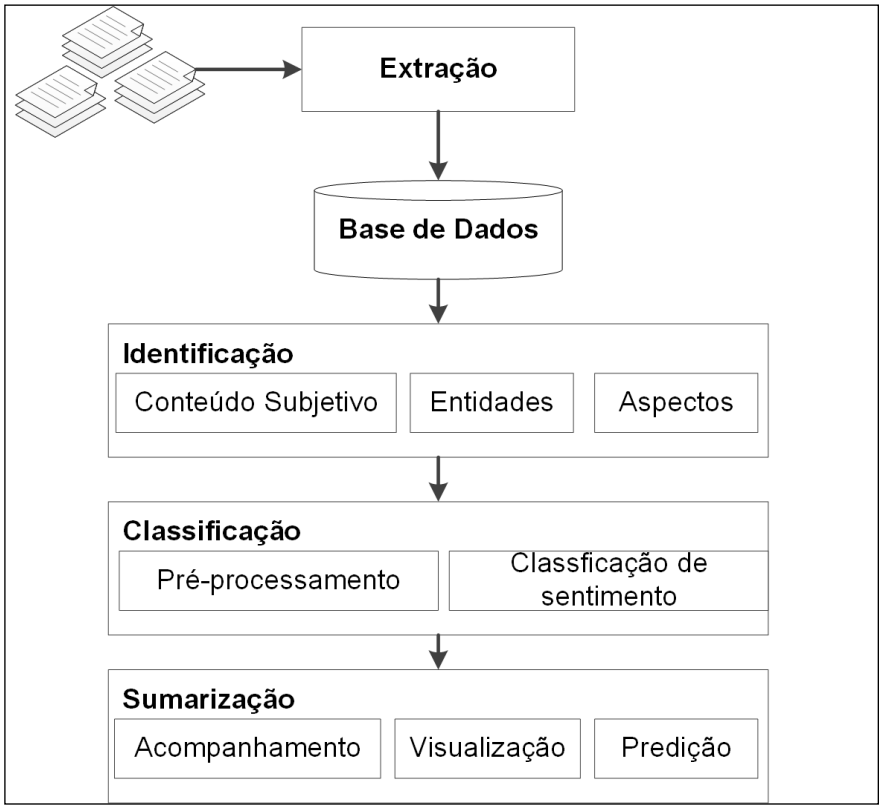
> A mineração de opiniões possui dois objetivos principais [47]: (i) identificar e discernir entre documentos que contêm fatos (notícias, por exemplo) e documentos que contêm opiniões, e (ii) classificar as opiniões quanto às suas polaridades, ou seja, se são opiniões positivas ou negativas (CARVALHO, 2014, p. 1).

As opiniões podem ser positivas, negativas ou neutras, indicando sua polaridade, sendo nos textos expressas por palavras opinativas, por exemplo: adjetivos = ruim, bom; advérbios = mal, devagar; e substantivos = amigos (SILVA; LIMA; BARROS, 2012).

Revolucionar a maneira que o computador é usado é uma motivação para estudos relacionados ao Processamento da Linguagem Natural (PLN). Computadores que conseguiram entender a linguagem natural puderam utilizar essa informação de forma rápida. A linguagem é um aspecto fundamental no comportamento humano. Na forma escrita, passa de geração para geração e na forma falada é um meio de comunicação entre pessoas (DA SILVA; MARTINS, 2008).

Como mostra a Figura 2, a mineração de opinião tem como características três tarefas: identificar, classificar e sumarizar (BECKER, 2014).

Figura 2 - Etapas da mineração de opinião



Fonte: Becker (2014).

A tarefa de identificação está associada à coleta de dados dos textos, localizando os tópicos do conteúdo a ser analisado. A tarefa de classificação da polaridade ou sentimento possui classificação binária, com classes positivas ou negativas, e há a classe neutra, quando não são encontradas polaridades claras nos textos minerados. Transformar os dados coletados e minerados em algo que o usuário entenda é a função da tarefa de sumarização, converter para métricas e sumários, com a finalidade de tornar os dados em números de diversidades de opiniões da entidade, métricas que mostram o potencial do sentimento nos dados.

O Aprendizado de Máquina (AM) possui algoritmos com o objetivo de melhorar o desempenho da capacidade do modelo de prever corretamente (métrica acurácia de classificação) a partir da experiência (ARRIAL, 2008). Esses algoritmos precisam de treinamentos prévios por meio de um conjunto de dados-treino de atributos quantitativos.

O algoritmo *Support Vector Machines* (SVM) chama atenção pelos resultados: possui muita assertividade, tem interpretação simples de situações não lineares complexas, sendo utilizado em relações lineares e não lineares, entre outros, tem aceitação em tarefas de classificação e predição. Existem muitas pesquisas no tempo de aprendizado, que é um dos problemas dessa técnica (CAMILO, 2009).

De modo geral, as redes sociais digitais se tornaram febre mundial. Falando-se em informação e comunicação entre pessoas, o acúmulo de informações postadas contendo opiniões é gigantesco. Com a internet, os usuários se sentem mais seguros para declarar suas opiniões referentes a algum acontecimento ou produto, situações em que cara a cara poderia haver certa intimidade (SANTOS, 2011). Nas redes sociais é que se encontram os maiores gargalos quando o assunto é interpretação

gramatical ou vocabulário, devido às abreviações, gírias, termos regionais, *emotions*, entre outros. São textos mais específicos e com opiniões inconstantes.

3 MATERIAIS E MÉTODOS

Este trabalho baseia-se no método estruturalista, que parte da investigação de um fenômeno concreto, indo em seguida para o nível abstrato, pelas normas de um modelo que represente o objeto de estudo, retornando por fim ao concreto, dessa vez com uma realidade estruturada. O método estruturalista caminha do concreto para o abstrato ou vice-versa (LAKATOS, 2010).

O Weka, sigla originada do nome *Waikato Environment for Knowledge Analysis* (Waikato Ambiente para Análise do Conhecimento), é um *software* bastante popular, sem custo. O pacote Weka está implementado na linguagem Java[4], que pode rodar em diversas plataformas e com linguagem orientada a objetos como modularidade, polimorfismo, encapsulamento, reutilização de código, dentre outros, sendo um *software* de domínio público (CIN, 2004).

O Weka possui uma coleção de algoritmos para aprendizado de máquina integrada, desenvolvida pela Universidade de Waikato (WAIKATO, 2015). Os algoritmos são integrados à ferramenta desenvolvida neste trabalho. As bibliotecas Weka podem ser chamadas por qualquer programa, facilitando a integração. A ferramenta Weka é desenvolvida em Java e publicada sob licença GPL.

No desenvolvimento da ferramenta que foi projetada neste trabalho, foram utilizadas diferentes tecnologias de acordo a necessidade da solução do objetivo. Na aplicação dos algoritmos de classificação e avaliação dos dados, será utilizada a API Java do Weka, a ferramenta do trabalho será desenvolvida através do Microsoft Visual Studio 2008, para fazer a ponte da linguagem Java (utilizada pelo Weka) e .NET, foi utilizada a *Java Virtual Machine* (JVM) a IKVM.NET[5], que tem a função de trazer as bibliotecas do Java para o .NET, utilizando-a na forma estática onde permiti que o código Java seja utilizado por aplicações .NET, etapa que o arquivo *.jar é convertido para *.dll.

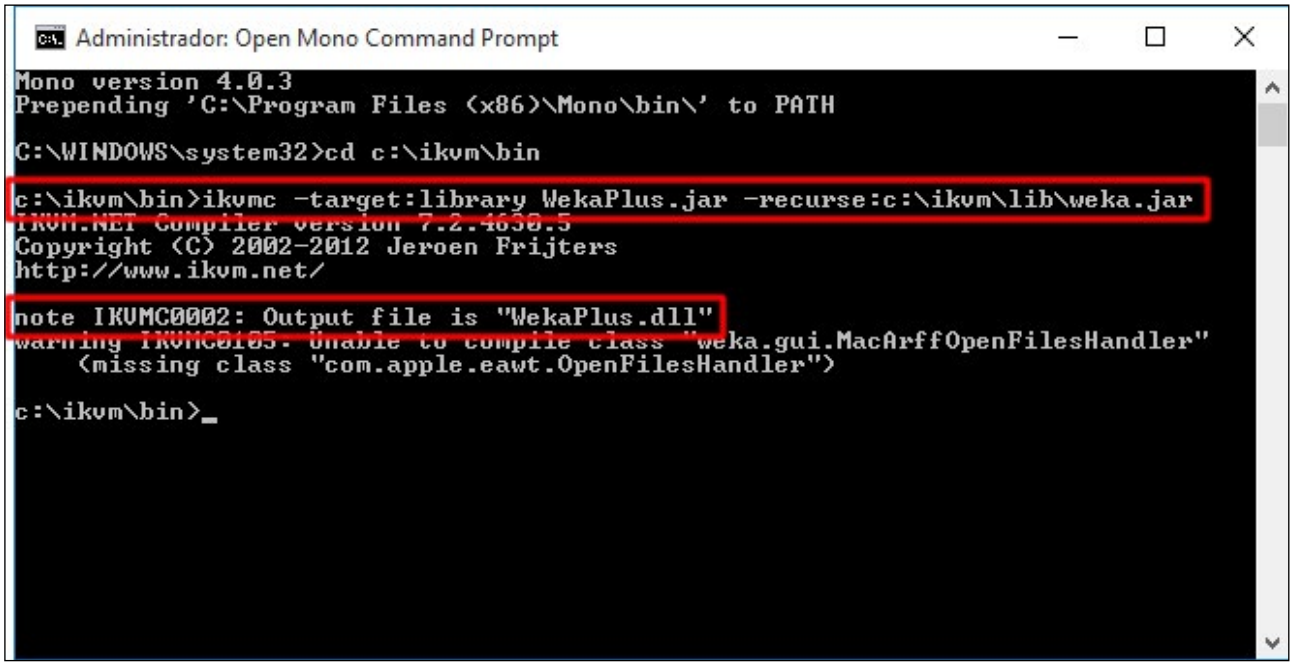
A versão Visual Studio 2008 foi escolhida devido a compatibilidade com a IKVM.NET. É dependente do projeto OpenJDK para implementação das bibliotecas do JDK até versão 7 onde havendo êxito de compilação (IKVM, 2015), tendo sido tentado utilizar o JDK 8 e não houve êxito na compatibilidade. Para utilização dos algoritmos do Weka no desenvolvimento do .NET, foi compilada a biblioteca Weka.jar. A compilação é executada pela plataforma Mono[6]. Segundo Mono (2015) a plataforma Mono é um *software* projetado para criação de aplicações multiplataformas. Por meio do Mono é chamado o ikvmc, que realiza a conversão *.jar para *.dll.

4 Disponível em: <https://www.java.com/pt_BR/>.

5 Disponível em: <http://www.ikvm.net/>.

6 Disponível em: <http://www.mono-project.com/>.

Figura 3 - Compilação WekaPlus.jar



Fonte: Elaborado pelos autores (2015).

O arquivo Weka.dll foi carregado nas referências do projeto AMD. Por meio disso a maioria das bibliotecas do Weka foram acessadas, mas não havendo êxito com a biblioteca "weka.classifiers. Evaluation", por motivo desconhecido. Essa biblioteca Evaluation é responsável pela avaliação dos dados classificados.

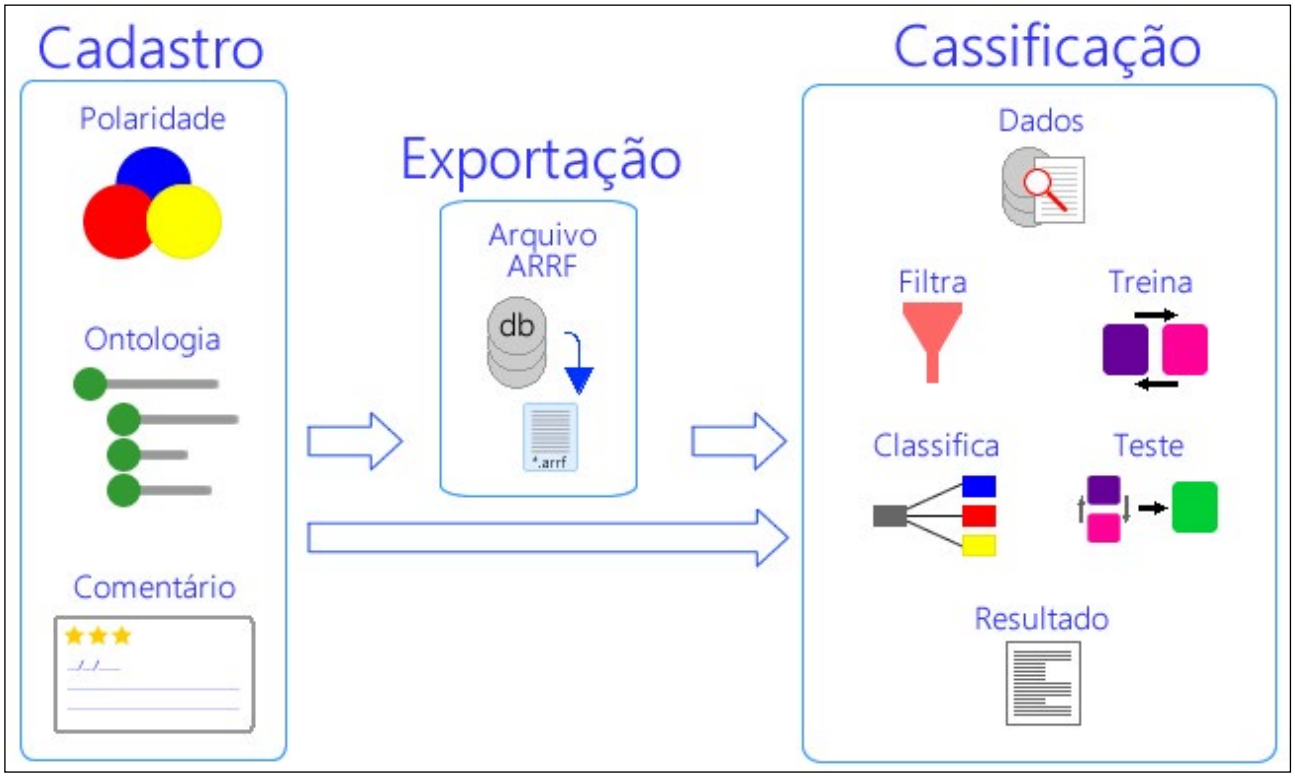
A avaliação dos dados é importante para a busca e amostra das classificações dos dados-treino. Tendo essa necessidade, foi criado um projeto WekaPlus na linguagem Java. Nesse projeto foi criada a biblioteca chamada EvaluationPlus.jar e adicionada a biblioteca Weka.jar, e nesse cenário foi possível acessar os recursos do Weka, tendo êxito na importação da biblioteca "weka. classifiers.Evaluation". Utilizando a ferramenta NetBeans IDE[7] como apoio no desenvolvimento do EvaluationPlus.jar, foram criadas as principais funções para utilização conforme necessidade para o projeto AMD.

4 RESULTADOS E DISCUSÃO

A ferramenta é composta por três módulos ou recursos: cadastros, exportação e classificação, como é possível visualizar na Figura 4.

7 Disponível em: <https://netbeans.org/>.

Figura 4 - Módulos/recursos da ferramenta



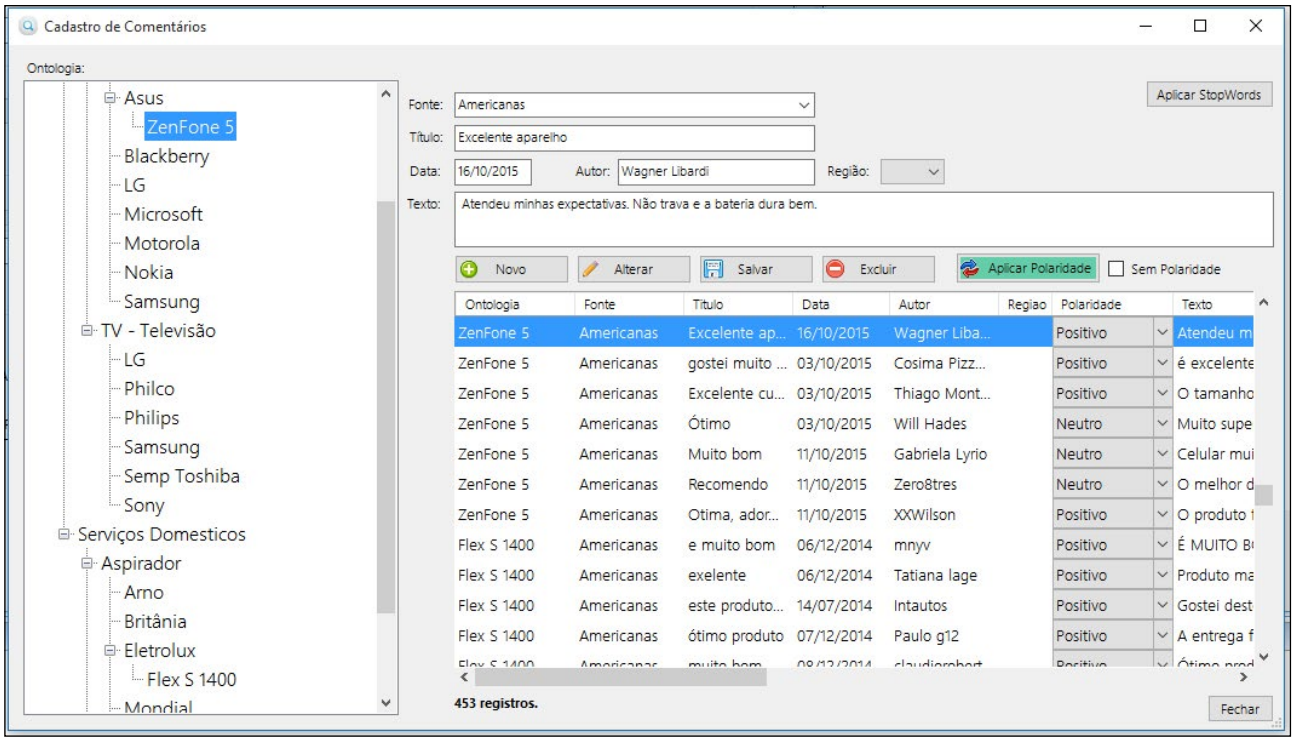
Fonte: Elaborado pelos autores (2015).

Na seção configuração/cadastro há possibilidade de configurar o acesso ao banco de dados. No cadastrado de ontologias (aspectos) é possível definir diversas ontologias pertinentes ao seu cenário de pesquisa. Conforme Suptitz (2013), para o apoio na classificação e recuperação de textos, existem trabalhos bem sucedidos na área de mineração de texto mostrando que o conceito de ontologia agrega em muito na obtenção de resultados. Um exemplo é a ontologia "Eletrônicos", na qual podem ser definidos termos-chave, conforme nível hierárquico, por exemplo: Smartphone / TV – Televisão / Apple / Asus / ZenFone 5 / Blackberry, entre outros.

As polaridades, como comentado na seção 2, são identificadas como positiva, neutra ou negativa, identificando a opinião/sentimento dos textos coletados, juntamente com as ontologias. O cadastro de polaridades possui o campo "Nome", no qual são atribuídos os nomes às polaridades, e um campo "Potencial Numérico", em que numericamente o usuário pode definir a qualidade dessa polaridade, por exemplo: ótimo= 2, bom= 1, neutro= 0, ruim= -1, péssimo= -2. A polaridade deve ser definida manualmente para que a ferramenta de mineração de dados tenha embasamento pela classe atribuída ao texto, assim, esses exemplos são usados na comparação com os dados que estão sendo minerados.

Na tela que exibe os textos, nos comentários cadastrados (FIGURA 5), o usuário realiza a manutenção dos dados coletados, associa as ontologias (aspectos) e sentimentos (polaridade), a fonte e informa outros dados como título, data, autor, região e texto. A opção "Aplicar StopWords" serve para remover as palavras que são irrelevantes, opção que é útil à classificação. Esse processo acessa um arquivo SW-PT.txt, localizado na raiz da ferramenta, que está carregado com 300 palavras em português, adquiridas no site (LINGUATECA.PT, 2015).

Figura 5 - Tela de cadastro de comentários



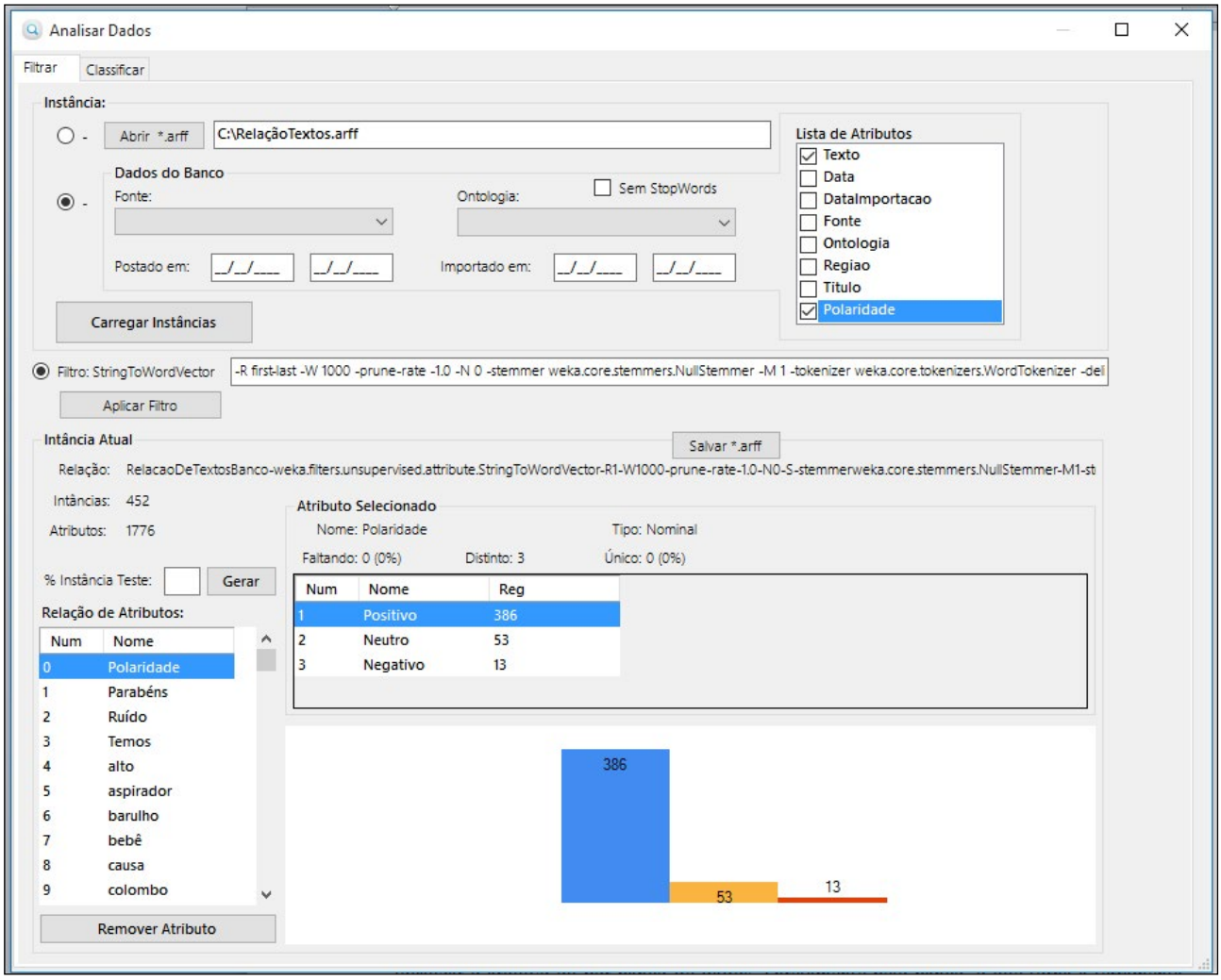
Fonte: Elaborado pelos autores (2015).

A opção "Aplicar Polaridade" atribui polaridade selecionada ao texto, no qual existe um combo na tabela relacionada conforme polaridades cadastradas. Como é um processo árduo ler todos os textos, analisar e atribuir um sentimento, o usuário informa o sentimento na coluna Polaridade, e aplica-a ao texto clicando na opção correspondente. Não há necessidade de a cada texto polarizado clicar nesse recurso. Há uma alternativa sem polaridade para filtrar somente os textos que ainda não possuem polaridade definida, auxiliando na aplicação.

Considerando que para aplicar os algoritmos de classificação é necessário gerar um arquivo no formato arff, o módulo de exportação foi desenvolvido na tela Gerar Arquivo arff. O usuário pode filtrar os dados de acordo com sua necessidade de análise. O arquivo arff é estruturado em duas partes: na primeira há lista com todos os atributos, em que definimos o tipo de atributo ou os valores que eles podem representar; na segunda parte constam os registros que serão minerados com os valores dos atributos para cada instância separados por vírgula. Caso não tenha algum registro, informa-se "?" (CIN, 2004).

No módulo classificação encontra-se a tela denominada "Analisar Dados", como mostra a Figura 6.

Figura 6 - Tela analisar dados - aba filtrar



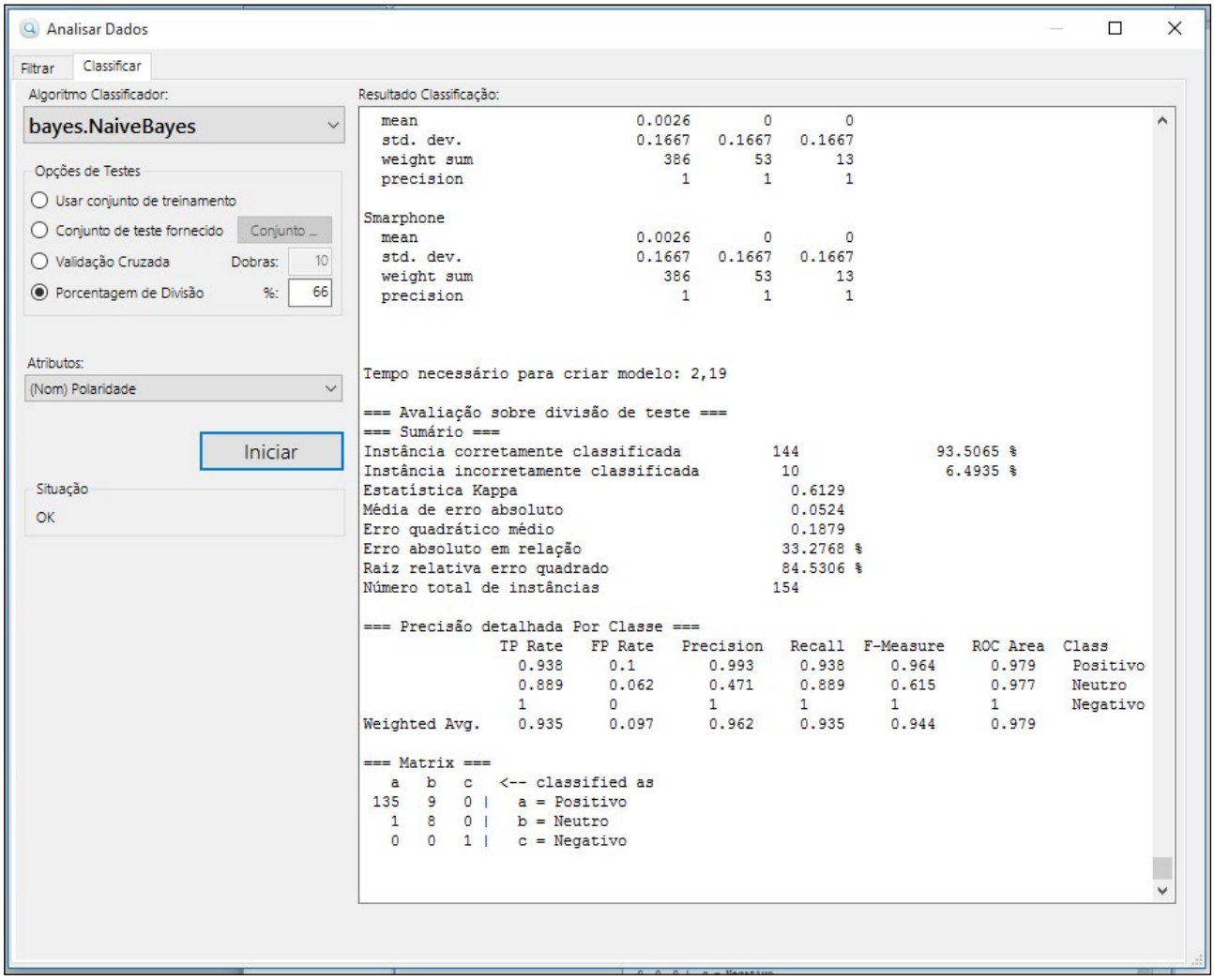
Fonte: Elaborado pelos autores (2015).

É possível efetuar a busca pelos dados a partir de duas opções: por meio de arquivos no formato arff ou diretamente pelo banco de dados. Esse recurso facilita a utilização do sistema e reduz a necessidade de conhecer o formato de arquivos que seguem o padrão arff.

Após carregar os dados, é possível remover atributos que não tenham importância para a análise usado a alternativa de remover atributos. Essa opção na ferramenta auxilia na etapa de pré-processamento da mineração dos dados, como comentado na seção 2, dando suporte ao analista no conjunto de dados, verificando valores válidos, preferências, restrições e evitando futuros problemas nos algoritmos. O filtro StringToWordVector escolhido possui diversas opções de filtragem, tendo o usuário liberdade de redefini-las do padrão.

Após a filtragem dos dados, é possível seguir para a classificação e análise dos dados, como mostra a Figura 7.

Figura 7 - Tela analisar dados – aba classificar



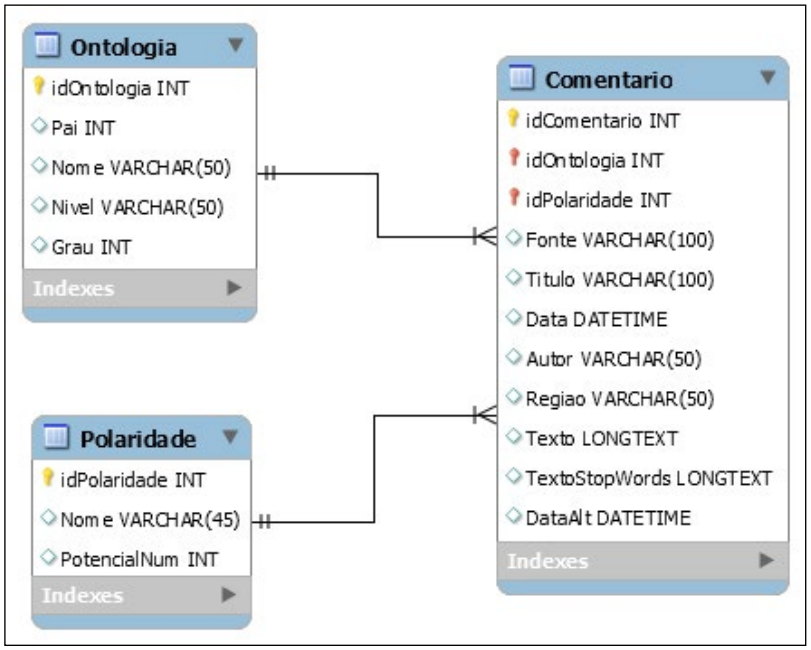
Fonte: Elaborado pelos autores (2015).

São disponibilizados diversos algoritmos para efetuar o treinamento e a classificação dos dados: NaiveBayes do grupo *bayes*, biblioteca "weka.classifiers.bayes.NaiveBayes"; SMO do grupo *functions*, biblioteca "weka.classifiers.functions.SMO" e NBTree do grupo *tree*, biblioteca "weka. classifiers.tree.NBTree".

Há quatro opções de testes: na opção "Usar conjunto de treinamento", "Conjunto de teste fornecido", "Validação Cruzada" e "Porcentagem de Divisão".

Para suportar os recursos do sistema, descritos anteriormente, foi elaborado o diagrama de entidade-relacionamento para o banco de dados seguindo o modelo relacional, como mostra a Figura 8. Esse banco de dados armazena as seções de configurações e classificação dos textos.

Figura 8 - Modelo banco de dados relacional



Fonte: Elaborado pelos autores (2015).

4.1 Testes executados

Após a ferramenta concluída, e com alguns dados coletados, ela começa a ser testada, sendo cadastradas três polaridades: positivo, neutro e negativo. A tabela Ontologia conta com 30 registros, dos quais, somente dois (Smartphone Zenfone 5 – Aspirador Flex S 1400) têm ligação direta com os comentários coletados. Foram coletados 251 comentários do Zenfone 5, um exemplo de comentário do cliente, e 201 comentários do Flex S 1400, totalizando 452 comentários cadastrados manualmente. Em seguida, todos os comentários foram também polarizados manualmente.

A Tabela 1 mostra o número de instâncias com suas polaridades conforme ontologia.

Tabela 1 - Tabela com o total de instâncias.

Polaridade	Instâncias ZenFone 5	Instâncias Flex S 1400
Positivo	230	156
Neutro	19	34
Negativo	2	11
Total	251	201

Fonte: Elaborado pelos autores (2015).

Para verificar a veracidade das avaliações, foi salva uma instância que já havia sido filtrada pelo StringToWordVector, utilizando os parâmetros padrões do filtro, para um arquivo arff, importando esse arquivo na ferramenta Weka e não realizando a filtragem.

Com os dados igualmente carregados, foi executada a classificação, com os resultados lado a lado das duas ferramentas, ambas com 452 instâncias e 1.851 atributos. A opção escolhida de teste foi a Validação Cruzada e o algoritmo NaiveBayes, onde se confirmou a compatibilidade nas avaliações.

Um dos problemas verificados é o tempo que o algoritmo leva para criar o modelo de aprendizagem de classificação. Com isso, outro teste realizado foi para verificar a eficiência de criação de modelos dos algoritmos, comparando a ferramenta desenvolvida com o *software* Weka. Os dados importados já haviam sido filtrados, havendo um total de 452 instâncias e 1.851 atributos, os quais receberam potências numéricas nessa classificação. Na Tabela 2, são mostrados os valores encontrados em segundos.

Tabela 2 - Eficiência criação modelos dos algoritmos

Algoritmo	Ferramenta desenvolvida	Weka
bayes.NaiveBayes	1,67 seg.	0,22 seg.
functions.SMO	1,26 seg.	0,14 seg.
trees.NBTree	1.312,61 seg.	95,32 seg.

Fonte: Elaborado pelos autores (2015).

O algoritmo NaiveBayes cria o modelo na em 1,67 segundos e no Weka leva somente 0,22 segundos, o algoritmo SMO na ferramenta desenvolvida cria em 1,26 segundos e no Weka leva somente 0,14 segundos, o algoritmo NBTree cria em 1.312,61 segundo e no Weka leva 95.32 segundos. Pode-se ver a grande perda de eficiência em tempo de criação, mas pensando-se em números maiores de dados, tende a ser um problema. O algoritmo NBTree é o que mais chama a atenção, tendo tempo mais significativo. Se convertermos os segundos para minutos, temos aproximadamente 22 minutos, tornando-se uma diferença muito expressiva.

Foi realizada avaliação utilizando a opção de teste "Usar conjunto de treinamento" com o algoritmo SMO, o qual não levou tempo significativo para realizar a avaliação nessa opção de teste. Vale destacar que das 452 instâncias, houve 100% de acerto na classificação das instâncias.

Outro resultado da opção de teste "Conjunto de teste fornecido", o arquivo (conjunto) origem é da mesma base anteriormente testada. Foram pegos 30% das instâncias e gerado um arquivo teste, e também não houve tempo significativo de espera. Das 135 instâncias testadas, houve índice de 92% de instâncias corretamente classificadas, valor muito considerável.

No resultado da "Validação Cruzada – 10 Dobras", na opção de teste ocorre a maior espera por resultados, devido a eles ocorrerem conforme o número de 10 dobras. Baseado nesses resultados, é feita média, tendo 82% de classificação correta.

Na opção de teste "Porcentagem de divisão – 66%", do total do conjunto de treinamento, 66% dos dados ficam treinos e os outros 34% ficam como teste; assim, os dados de treinamento são aplicados nesses 34% e o tempo de avaliação se mostrou insignificante para a demanda. Vale destacar que nas 154 instâncias avaliadas, houve 100% de instâncias corretamente classificadas.

5 CONCLUSÃO

A conclusão da ferramenta mostra que o objetivo principal deste trabalho foi alcançado. A definição dos aspectos nos textos é realizada por meio do cadastro de ontologia e o sentimento pelo cadastro de polaridade. A utilização de recursos da ferramenta Weka para realizar o processo de mineração de dados foi alcançada por meio de documentações disponibilizadas na *web*.

Os recursos disponibilizados visam a facilitar o processo de análise de textos, a definição de aspectos e alvos do sentimento contido nos textos, além de automatizar a conversão dos dados no formato necessário para aplicação dos algoritmos. Boa parte dos *softwares* para mineração de dados exige que o usuário realize essas tarefas de forma manual e que ele tenha conhecimentos sobre estruturas de arquivos e algoritmos de aprendizagem de máquina.

Uma das limitações percebidas diz respeito ao tempo de processamento, que se mostrou superior a outros *softwares*, principalmente o Weka, ao qual foi comparado. Esses tempos podem ser reduzidos pela aplicação de técnicas de programação paralela ou a partir de melhorias na arquitetura da ferramenta desenvolvida.

Até o presente momento o sistema realiza parte das funcionalidades esperadas de uma ferramenta para classificação automática. São testados os algoritmos de aprendizado de máquina e é possível definir aspectos associados a textos, entretanto, em trabalhos futuros, espera-se manter de forma permanente os resultados do aprendizado e aplicar os algoritmos a novas publicações.

REFERÊNCIAS

ARRIAL, Roberto Ternes. **Predição de RNAs não-codificadores no transcriptoma do fungo Paracoccidioides brasiliensis usando aprendizagem de máquina**. 2008. Tese de Doutorado. Instituto de Biologia.

BECKER, Karin. Introdução à Mineração de Opiniões. **XXXIV Congresso da Sociedade Brasileira de Computação.** Porto Alegre, RS, cap. 4, pág. 125-176, 2014.

CAMILO, Cássio Oliveira; SILVA, João C. **Mineração de dados: Conceitos, tarefas, métodos e ferramentas.** Universidade Federal de Goiás (UFC), p. 1-29, 2009.

CARVALHO, Jonnathan dos Santos. **Uma estratégia estatística e evolutiva para Mineração de Opiniões em *Tweets*.** 2014.

CIN. Weka.doc. **Centro de Informática de UFPE - CIn**, Pernambuco, 23 Abril, 2004. Disponível em: <www.cin.ufpe.br/~mcps/IA/IA2004.1/weka.doc>. Acesso em: 03 jun. 2015.

DA SILVA, Wilson Carlos; MARTINS, Luiz Eduardo Galvão. **PARADIGMA: Uma Ferramenta de Apoio à Elicitação e Modelagem de Requisitos Baseada em Processamento de Linguagem Natural**. WER, v. 8, p. 140-151, 2008.

DIAS, Maria Abadia Lacerda; DE GOMENSORO MALHEIROS, Marcelo. **Extração Automática de Palavras-chave de Textos da Língua Portuguesa.** Centro Universitário UNIVATES, 2005.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37, 1996.

IKVM. **IKVM.NET - User's_Guide**. Disponível em: <*http://sourceforge.net/p/ikvm/wiki/User's_Guide/*>. Acesso em: 23 out. 2015.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 7. ed. São Paulo: Atlas, 2010.

LINGUATECA.PT. **Listas de palavras frequentes em português, *Lists of Portuguese stopwords***. Disponível em: <http://www.linguateca.pt/chave/stopwords/folha.MF300.txt>. Acesso em: 07 out. 2015.

MONO. **Mono Project**. Disponível em: <http://www.mono-project.com/>. Acesso em: 23 out. 2015.

REZENDE, S. O. et al., **Mineração de Dados, in REZENDE,** S. O. (Eds.), Sistemas Inteligentes, Editora Manole Ltda., p.307-335. 2003.

SANTOS, Leandro Matioli et al. Twitter, análise de sentimento e desenvolvimento de produtos: Quanto os usuários estão expressando suas opiniões?. **Revista PRISMA.COM**, n. 13, 2011.

SILVA, Nelson Rocha; LIMA, Diego; BARROS, Flávia. SAPair: Um Processo de Análise de Sentimento no Nível de Característica. In: **4nd International Workshop on Web and Text Intelligence (WTI'12)**, Curitiba. 2012.

SUPTITZ, Ivan Luis. **Aplicação de Mineração de Texto com o apoio de Ontologias para extração de conhecimento em Bases de Dados Textuais.** UNISC, Santa Cruz do Sul, RS, 2013.

WAIKATO. Weka Knowledge Explorer. **The University of WAIKATO.** Disponível em: <http://www.cs.waikato.ac.nz/~ml/weka/gui_explorer.html>. Acesso em: 03 jun. 2015.