UnderMine Text Miner – Uma Ferramenta de Mineração de Texto para Área Educacional

Carine G. Webber – CCTI/UCS – cgwebber@ucs.br

Lauren Girardi Cristofoli – CCTI/UCS – lgcristo@ucs.br

Maria de Fátima Webber do Prado Lima – CCTI/UCS – mfwplima@ucs.br

Resumo. *Este artigo apresenta um sistema de mineração de textos, denominado UnderMine Text Miner, que utiliza algoritmos embasados nas metodologias imunológicas. O UnderMine Text Miner opera basicamente dois processos. No primeiro processo de treinamento, o sistema é executado sucessivamente a fim de apreender palavras que contextualizam textos lidos para cada área de estudo informada pelo usuário. No segundo processo de classificação, o sistema analisa um novo texto e o classifica segundo as áreas de estudo aprendidas no primeiro processo. Para finalizar, o artigo descreve experimentos preliminares realizados e exemplos.*

Palavras chave*: Mineração de Textos, Sistemas Imunológicos Artificiais, Classificação de textos.*

UnderMine Text Miner - A Text Mining Tool for Educational Area

Abstract. *This article presents a text mining system, named UnderMine Text Miner, that uses immunologic algorithms. Basically, the UnderMine Text Miner has two processes. In the first training process, the system is trained in order to learn words that contextualize text in each study area informed by the user. In the second classification process, the system analyzes a new text and classifies it according to study area learned in training process. To conclude, this article describes previous experiments and examples.*

Keywords*: Text Mining, Artificial Immune Systems, Text classification.*

1. Introdução

A Mineração de Dados Educacionais (MDE) é uma área de pesquisa que desenvolve métodos para extrair dados oriundos de ambientes educacionais a fim de descobrir padrões ou evidências científicas sobre estudantes e formas de aprendizagem. As técnicas diferem frequentemente das técnicas da mineração de dados tradicional, pois devem explorar níveis de hierarquia e organização dos dados educacionais. A MDE utiliza um ciclo interativo na formação de hipóteses, nos testes e nos refinamentos necessários. Para realizar a análise dos dados diversos métodos são utilizados pela MDE: predição, agrupamento, descoberta de relações, mineração de textos (MT) e análise de redes sociais (Sachin, 2012). A MT é uma subárea que procura descobrir, de forma automática, informações (padrões e anomalias) em dados não estruturados, sendo

a maioria em formato textual.

Os paradigmas atuais de programação enfrentam grandes dificuldades na interpretação de textos pelo fato de que muitas vezes palavras podem ter relações difusas e/ou significados ambíguos (Hotho, 2005). Além disso, a pesquisa por termos relevantes em um texto é um trabalho árduo, pois uma palavra não pode ser analisada separadamente. É necessário considerar o contexto no qual ela se encontra e respeitar regras ortográficas e sintáticas que condizem ao idioma do texto que está sendo analisado. Até o momento não existem algoritmos ótimos que consigam lidar com tais dificuldades de forma satisfatória.

Quando a MT é utilizada na MDE, a aplicação de algoritmos eficientes e adequados é primordial. O desenvolvimento da ferramenta *UnderMine Text Miner* visa contribuir com esta área de pesquisa, apresentando uma nova solução computacional baseada em Sistemas Imunológicos Artificiais (SIA). Neste contexto este artigo está organizado em 6 seções. A seção 2 introduz os principais conceitos associados a MT e aos SIA. A seção 3 descreve alguns trabalhos desenvolvidos na área de MT. A seção 4 aborda o método de pesquisa utilizado no experimento desenvolvido. Finalmente, a seção 5 apresenta os resultados obtidos no experimento e a seção 6 conclui o artigo.

2. Mineração de Dados e Sistemas Imunológicos

O processo de MT pode ser dividido em cinco etapas (Aranha, 2007). A coleta de dados é a primeira etapa e tem como função formar uma base de dados textual. A segunda etapa, o pré-processamento, visa estruturar os documentos, organizando-os, formatando-os e, consequentemente, melhorando sua qualidade para as etapas seguintes. É nesta etapa que as *stopwords* são retiradas, o texto é dividido em palavras e estas são classificadas de acordo com a classe gramatical. A seguir, na etapa de mineração é realizada a aplicação de algoritmos sobre os documentos com o intuito de extrair conhecimento. Na última etapa é feita a interpretação dos resultados obtidos.

Na MT, os termos são selecionados conforme a sua frequência no documento. Durante a computação do cálculo de frequência, cada vez que uma palavra é encontrada é atribuído um valor ao relacionamento da palavra com o texto, denominado peso. Este valor indica a importância que a palavra exerce sobre o texto. Os valores de peso variam de zero (termos com pouca importância) a um (maior importância). Existem diversos métodos para a aplicação destes cálculos, tais como: frequência absoluta, frequência relativa e frequência inversa de documentos (Morais, 2007). Dentre os modelos mais utilizados, pode-se destacar o modelo booleano, o espaço-vetorial, o probabilístico, o difuso e o aglomerado (Ebecken, 2003; Wives, 2002).

Um SIA pode ser utilizado no processo de MT para auxiliar na aprendizagem do texto, qualificando a descoberta de informações. SIA é uma área de estudo da Inteligência Artificial composta por metodologias inteligentes baseada no funcionamento do sistema imunológico dos seres vertebrados para a solução de problemas reais (Dasgupta, 1999). Desde a década de 90, os SIA vêm sendo utilizados para reconhecimento de padrões, falhas de segurança e mineração de dados (De Castro, 2001). O uso destes sistemas pode ser destacado pelos aspectos de unicidade (onde cada ser possui seu próprio sistema imunológico) com suas particularidades; reconhecimento de padrões internos e externos ao sistema, detecção de anomalias, detecção imperfeita (tolerância a ruídos), e a diversidade que o SI oferece, com uma quantidade limitada de células e moléculas que são utilizadas na obtenção do reconhecimento de elementos. Na aprendizagem por reforço o SIA melhora sua resposta a cada encontro com o mesmo patógeno e, por fim, ele desenvolve uma memória, visando uma resposta mais efetiva (os componentes que reconhecem e combatem as patologias de forma satisfatória são

armazenados). O SIA disponibiliza diversos algoritmos utilizados na computação, tais como: seleção negativa, expansão e seleção clonal e redes imunes.

3. Trabalhos Relacionados

Vários trabalhos têm descrito novas propostas de algoritmos e ferramentas de MT em diferentes áreas do conhecimento. Yang (2012) desenvolveu um procedimento automático para descobrir eventos e assuntos úteis sobre inteligência estratégica. Primeiro é aplicada uma técnica de agrupamento nos dados treinados para obter os relacionamentos entre os documentos. Após, um processo da detecção da inteligência é aplicado ao resultado do agrupamento para descobrir os dados estratégicos. Ma (2012) apresentou um método para agrupar propostas de pesquisas de acordo com suas similaridades. Esta classificação automática auxiliaria agências de governo e instituições de pesquisa privadas agrupar as propostas de projeto recebidas. A solução proposta foi desenvolvida utilizando o conceito de sistemas de ontologia. Yi (2012) desenvolveu um método para realizar a MT que utiliza dois algoritmos de aprendizagem da máquina: rede neural *Back-Propagation* e o algoritmo do vizinho mais próximo.

Diversas propostas têm sido desenvolvidas para atender demandas dentro da área educacional. Hsu (2012) utilizou a MT para analisar os diários de classe dos professores, onde pode-se extrair diferentes tipos de informação: conteúdos trabalhados, princípios pedagógicos utilizados, conhecimento do currículo, conhecimento dos estudantes e suas características, conhecimentos dos fins educacionais, propostas e valores. Pushpalatha (2012) cita a importância para a área educacional da descoberta rápida de documentos relevantes, apresentando uma proposta onde as medidas baseadas em peso e frequência deveriam ser substituídas por valores da discriminação do termo.

Lupi (2012) desenvolveu um protótipo, denominado Diy Gis, para ensinar os principais elementos do planejamento, projeto e gerenciamento de recursos urbanos. Entre outros métodos, a Diy Gis utiliza MT e análise de conversação para localizar conteúdos gerados por usuários a fim de obter mapas e imagens significativas. Reategui (2012) desenvolveu uma ferramenta (*Sobek*) para extrair gráficos de texto, auxiliando os estudantes desenvolverem resumos de textos. A ferramenta foi desenvolvida utilizando um algoritmo baseado no modelo de distâncias, onde os nós representam os principais termos encontrados no texto e as bordas definem as informações secundárias. Peng (2012) desenvolveu uma aplicação, denominada *iSimp*, para reconhecer textos biomédicos, simplificando-os, a fim de extrair as informações mais relevantes. A ferramenta *iSimp* utiliza a análise léxica ao invés da análise sintática para realizar o reconhecimento do texto. Houjeij (2012) apresentou um novo método que analisa o emocional do discurso humano, considerando o áudio e as produções textuais. Nesta proposta, os dados do texto foram analisados utilizando a máquina de suporte vetorial e o algoritmo K-Média. Já os dados do discurso foram analisados através de lógica *fuzzy* e do algoritmo do vizinho mais próximo.

Pode-se verificar que existem vários trabalhos sendo realizados para aprimorar as técnicas de MT. Uma das abordagens de pesquisa para isso se vale dos SIA. As seções seguintes apresentam uma ferramenta de MT baseada em conceitos dos SIA.

4. Sistema UnderMine Text Miner

O sistema *UnderMine Text Miner* opera em duas etapas: Treinamento, encarregada da aprendizagem dos *tokens* de artigos representativos dos domínios e a etapa de Classificação, que efetua a MT propriamente dita e classifica um novo artigo.

4.1 Treinamento dos Artigos

O objetivo do sistema de treinamento é ler cada um dos artigos e selecionar os termos que possuem maior relevância para cada documento, calculando sua frequência e um limiar que será posteriormente utilizado para a sugestão da classificação de um novo artigo em relação à sua área de estudo. Para realizar esta etapa, os conceitos de Teoria do Perigo dos SIA são utilizados.

Os artigos utilizados para o treinamento estão agrupados por área de estudo e o sistema busca esses artigos na pasta selecionada pelo usuário. Ao iniciar o treinamento, é criada uma lista contendo os artigos lidos. Cada artigo é analisado e seus termos são separados um a um. Cria-se então uma nova lista que receberá esses termos, agora denominados *tokens*. Cada *token* carregará consigo a sua frequência, ou seja, o número de vezes que aparece em todas as mensagens do mesmo tipo e será armazenado em um arquivo texto junto com os demais t*okens*. Como o treinamento é feito individualmente para cada área de estudo, são criados diversos arquivos texto, cada um relacionado a uma área com seus respectivos *tokens.* A tabela 1 exibe os 10 termos mais frequente de duas áreas de estudo gerados pelo sistema de treinamento.

Com o intuito de auxiliar na fase de classificação é calculada a concentração dos termos de maior frequência. Esta propriedade consiste no somatório do número de ocorrências que o termo possui em cada um dos artigos que compõem o corpus. Após, é realizado o cálculo que determina o limiar do *token*. Este limiar indica a proximidade que o artigo possui com cada uma das áreas de estudo aprendidas no treinamento, o que facilita a visualização dos resultados e a classificação do artigo por parte do usuário.

Tabela 1. Termos gerados pelo sistema de treinamento

Termos História	Termos Biologia
106|WAR|HISTORY	40|CELLS|BIOLOGY
31|WARS|HISTORY	29|CELL|BIOLOGY
28|CIVIL|HISTORY	29|PROTEIN|BIOLOGY
18|SUPPORT|HISTORY	26|APOPTOSIS|BIOLOGY
16|DOMESTIC|HISTORY	26|ER|BIOLOGY
16|PEACE|HISTORY	25|STRESS|BIOLOGY
16|MILITARY|HISTORY	22|INDUCED|BIOLOGY
15|CONFLICT|HISTORY	21|CANCER|BIOLOGY
13|INTERSTATE|HISTORY	20|C|BIOLOGY
13|DEMOCRACY|HISTORY	20|FAD|BIOLOGY

A fórmula do cálculo das concentrações corresponde à soma das frequências dos termos sobre o número total de termos em comum entre o treinamento e o mesmo artigo. A interface do sistema de treinamento (figura 1) permite que o usuário visualize quais os termos mais frequentes no conjunto de artigos lidos para a área de estudo determinada, bem como a concentração que esses termos possuem em cada artigo. O processo de treinamento é iniciado ao clicar no botão "*Processing article*", que faz a leitura sequencial de cada um dos artigos existentes na pasta selecionada pelo usuário e o seu cálculo de frequência. Ao clicar no botão "*Concentration article*" o sistema faz o cálculo de concentração e exibe o valor obtido para cada artigo, o total de artigos lidos, a média das concentrações destes artigos e o desvio padrão dos mesmos.

O sistema de treinamento foi implementado na linguagem C#, estruturado em três camadas (Interface, Treinamento e Representações), possuindo as seguintes classes: Treinamento, Mensagem, *Token*, Arquivo, Artigo e *StopWord*. Na camada de interface há a classe Treinamento, encarregada da comunicação com as outras classes do sistema. Na camada de treinamento há a classe de representação dos artigos e as classes referentes à manipulação de arquivos e *strings*. Por fim, na camada de representações

ficam as classes *Token* e Mensagem. A classe Artigo é responsável pelo treinamento dos artigos lidos pelo sistema. Nela realizam-se os cálculos de frequência e concentração dos termos no artigo. A manipulação dos arquivos é feita através dos métodos existentes na classe Arquivo. É a partir dela que os artigos são lidos em seus respectivos diretórios e os termos são separados em arquivos texto.

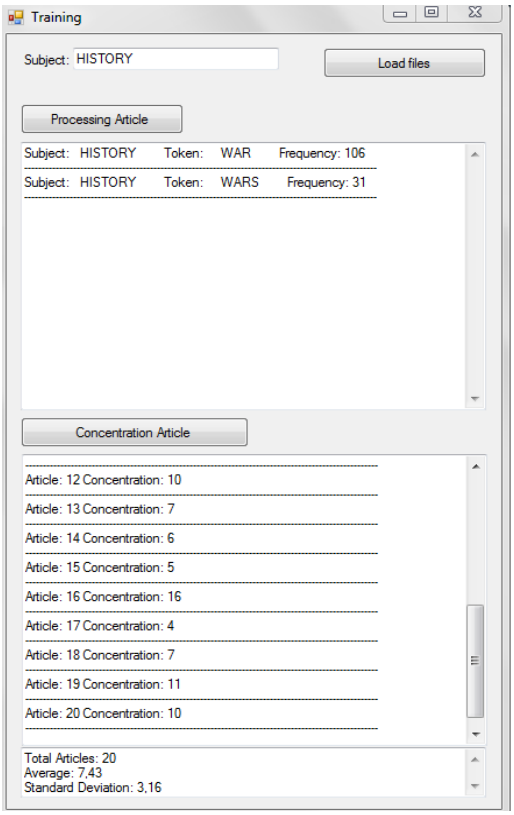

Figura 1. Interface sistema de treinamento

A classe Mensagem representa os atributos de um objeto do tipo mensagem, denominados id e concentração; possui também os atributos de uma lista de objetos do tipo mensagem que são média e desvio padrão. Os *tokens* são representados pela classe *Token*, que carrega para cada objeto desse tipo os atributos nome, tema de estudo e número de ocorrências do *token*. Por fim, a classe *StopWord* fica responsável pela remoção das *stop words* encontradas no artigo. Consideram-se *stop words* as palavras que se tornam irrelevantes para análise do contexto de um conjunto de dados, por isso são removidas.

4.2 Classificação de Artigos

O sistema de classificação de artigos científicos se baseia na metodologia imunológica proposta, onde o sistema de treinamento cria os anticorpos (*tokens* de cada domínio) e o sistema de classificação submete o antígeno (artigo bruto) para o diagnóstico. Na interface do sistema (Figura 2), o usuário insere um artigo em sua forma bruta e a classificação é iniciada ao clicar no botão "*Start diagnosis*". O botão "*Clear*" limpa as informações dos campos texto e permite a realização de um novo diagnóstico.

Ao iniciar a leitura do artigo inserido, o *UnderMine* separa cada *token* e verifica se este já foi aprendido, armazenando esta informação em uma lista. No final da análise, a ferramenta apresenta no quadro "*Diagnosis Results*" a lista de *tokens* obtida na etapa anterior e a concentração que o artigo possui para cada domínio. Um gráfico de pizza

apresenta a proximidade do artigo com cada domínio através de percentuais.

O sistema de classificação de artigos foi desenvolvido na linguagem C# com uma arquitetura em 3 camadas: camada de interface, dados e verificadores. A camada de interface possui a interface do sistema e uma classe que faz a comunicação entre as outras camadas. As classes que fazem a manipulação de dados do sistema ficam definidas na camada de dados. Por fim, a terceira camada contém os verificadores SIA. A tabela 2 representa as camadas do sistema e suas respectivas classes.

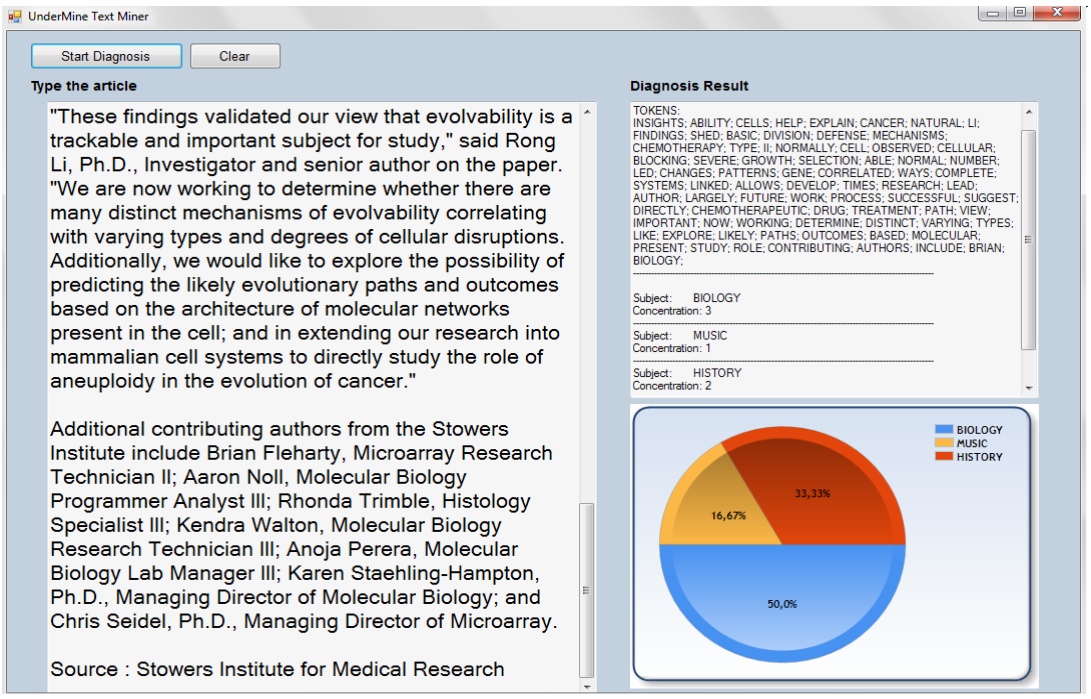

Figura 2. Interface do sistema de diagnóstico

Na camada de interface foram implementadas duas classes: UnderMine e Diagnosticar. A classe *UnderMine* é a interface do sistema, onde são definidas as validações de interface, os métodos que escrevem o diagnóstico na tela e a chamada para a classe Diagnosticar. A classe Diagnosticar, por sua vez, realiza a comunicação com as outras camadas do sistema.

Tabela 2. Representação das camadas do Sistema de Classificação

Interface		Dados		Verificadores
UnderMine	Interface do sistema.	CarregarDados	Leitura dos arquivos que contêm os termos aprendidos no treinamento.	AgenteDecompositor
Diagnosticar	Realiza a comunicação com outras camadas do sistema.	QuadroNegro	Possui listas com os termos lidos na CarregarDados. Faz a comunicação entre os agentes do sistema.	VerificadorConcentracaoTokens
		Termos	Atributos dos objetos do tipo termo.	
		Token	Atributos dos objetos do tipo token.	

A camada de dados contém as classes CarregarDados, QuadroNegro, Termos, *Token* e ResultadoDiagnostico. A classe CarregarDados fica encarregada da leitura dos arquivos que contêm os termos aprendidos para as áreas de estudo. Estes termos são lidos e adicionados em listas no quadro negro. A classe QuadroNegro faz a comunicação entre os agentes do sistema, que adicionam novos atributos a essa classe sempre que necessitam acessar uma nova informação. As classes Termos e *Token* representam atributos dos objetos do tipo termos e *token*, respectivamente.

A camada de verificadores SIA é a principal camada do sistema, onde estão implementados todos os agentes que representam as verificações do sistema imune. Cada agente foi definido por uma *thread* no sistema que é inicializada no início do diagnóstico e fica encarregada do monitoramento do ambiente, ou seja, se a *thread* detectar uma informação necessária ela irá executar suas tarefas. Do contrário, ela permanece em estado de espera apenas monitorando o ambiente.

Cada agente possui os estados ativo, inativo e concluído. Se o agente estiver em estado inativo é porque o mesmo está monitorando o ambiente e aguardando o momento de iniciar suas tarefas; se ele estiver executando suas tarefas seu estado é ativo e, ao finalizá-las, seu estado passa a ser concluído. A camada de verificadores possui duas classes: AgenteDecompositor e VerificadorConcentracaoTokens. A classe AgenteDecompositor captura o artigo inserido pelo usuário e o decompõe em *tokens* que são armazenados em uma lista. Essa lista de *tokens* é comparada com a lista de *tokens* obtidos no treinamento, tendo como produto uma terceira lista contendo os termos em comum. A classe VerificadorConcentracaoTokens é o agente responsável por identificar e analisar as variáveis definidas pelo agente da resposta imune inata. Ela calcula a concentração e limiar de termos para retornar o diagnóstico.

5. Resultados e Discussão

Foi definido um cenário de testes para o sistema implementado com o intuito de validar os resultados retornados por este. Para dar início ao diagnóstico clica-se no botão "*Start Diagnosis*". O resultado retornado pela execução do sistema é escrito em "*Diagnosis Result*", onde são exibidos os *tokens* mais relevantes do artigo analisado e a concentração que estes *tokens* possuem em comparação com cada área aprendida no treinamento. Na interface de diagnóstico (figura 3), o resultado é reiterado através de um gráfico que apresenta através de percentuais a proximidade que o artigo possui com cada área de estudo comparada.

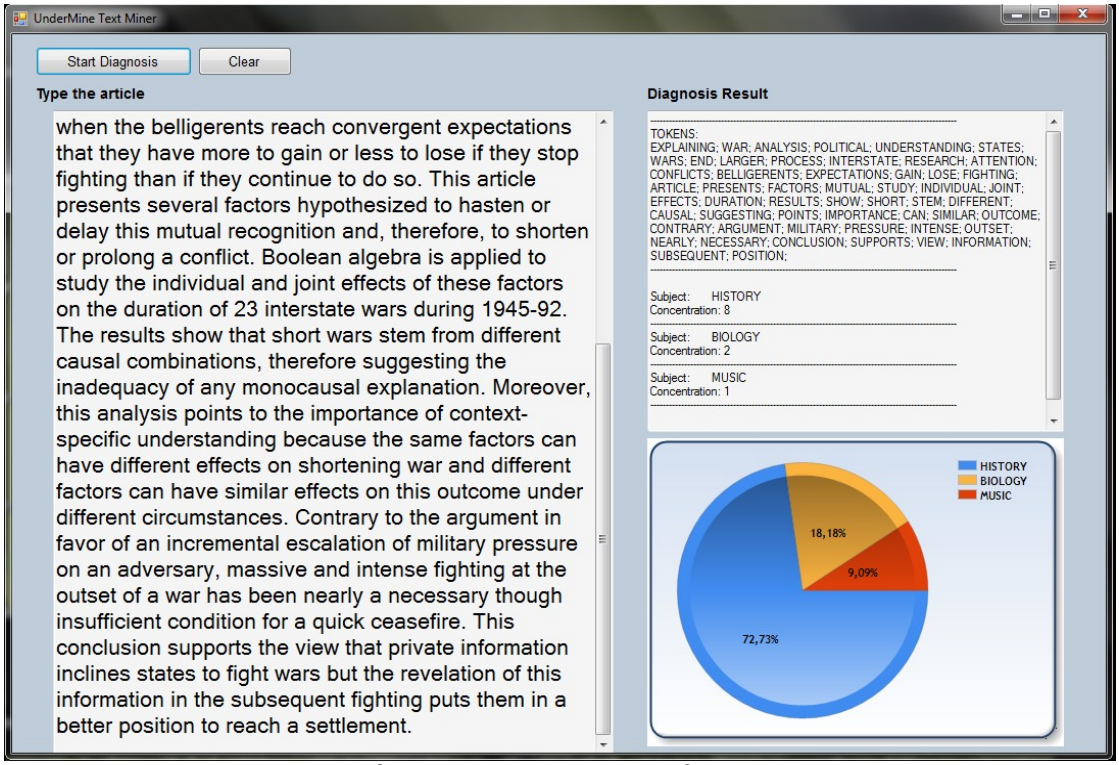

Figura 3. Classificação de um artigo na ferramenta *UnderMine*

Neste cenário de testes foram inseridos 10 artigos de História, referentes ao assunto de guerras civis; 10 artigos de Biologia, relacionados a estudos celulares e 10 artigos aleatórios dentro da área de Música. A tabela 3 apresenta os resultados obtidos.

Tabela 3. Resultados retornados nos testes da ferramenta UnderMine

ID Texto	Classificação correta	Classificação obtida	Resultado obtido
B1	Biologia	Biologia	Correto
B2	Biologia	Biologia	Correto
ID Texto	**Classificação correta**	**Classificação obtida**	**Resultado obtido**
B3	Biologia	Biologia	Correto
B4	Biologia	Biologia	Correto
B5	Biologia	Biologia	Correto
B6	Biologia	Biologia	Correto
B7	Biologia	Biologia	Correto
B8	Biologia	Biologia	Correto
B9	Biologia	Biologia	Correto
B10	Biologia	Biologia	Correto
H1	História	História	Correto
H2	História	História	Correto
H3	História	História	Correto
H4	História	História	Correto
H5	História	História	Correto
H6	História	História	Correto
H7	História	História	Correto
H8	História	Indefinida	Incorreto
H9	História	História	Correto
H10	História	Indefinida	Incorreto
M1	Música	Música	Correto
M2	Música	Música	Correto
M3	Música	Música	Correto
M4	Música	Música	Correto
M5	Música	Música	Correto
M6	Música	Música	Correto
M7	Música	Música	Correto
M8	Música	Música	Correto
M9	Música	História	Correto
M10	Música	Música	Correto

O sistema obteve 93,3% de acerto ao indicar o domínio ao qual o artigo inserido pertence. Os resultados apresentados pela classificação são satisfatórios e precisos, pois são obtidos através de cálculos de concentração com o auxílio de limiares, retornando ao usuário todas as possibilidades de classificação cabíveis ao teor do documento.

5.1 Análise Comparativa dos Resultados

A ferramenta *Weka* agrupa uma coleção de algoritmos de aprendizagem de máquina com a funcionalidade de realização de tarefas referentes à mineração de dados (Witten e Frank, 2000). Utilizou-se o *Weka* como ambiente de testes para comparação com a ferramenta *UnderMine*. Dentre todos os algoritmos disponíveis no *Weka*, optou-se pela utilização do algoritmo SMO (Sequential Minimal Optimization), uma implementação de máquinas de suporte vetorial. Esta técnica obtém excelentes resultados em MT.

Para a realização dos testes foi elaborado um *dataset* composto pelos termos mais frequentes das áreas de estudo. Isto foi necessário pois o Weka não trabalha com textos diretamente. O *dataset* constitui-se por um arquivo do tipo .arff que contém uma matriz estruturada de tal forma que cada coluna da matriz representa um atributo e cada linha um artigo. Cada atributo é um termo extraído dos arquivos de treinamento.

Em suma, o *dataset* contém 20 atributos (colunas), sendo os 10 primeiros os termos mais frequentes identificados no treinamento de História e os 10 últimos os termos mais frequentes do conjunto de artigos de Biologia. No total há 40 instâncias: as

10 primeiras correspondem aos artigos utilizados no corpus de História, as 10 seguintes pertencem ao corpus de Biologia; as próximas 10 instâncias são artigos utilizados na base de testes de História e as 10 restantes são utilizadas na base de testes de Biologia.

Os atributos são binários, onde zero indica que o artigo contém o termo e um indica que o arquivo não contém o termo em questão. O último atributo é nominal e caracteriza a categoria do artigo, podendo ser do tipo "história" ou "biologia".

5.2 Resultados *Weka*

A execução do algoritmo SMO foi feita utilizando as opções *default* do ambiente e a opção de *cross-validation*. Esta opção opera dividindo o conjunto total de dados em subconjuntos, separando um deles para testes e o restante para estimação dos parâmetros e cálculo de acurácia. Este processo é realizado várias vezes, alterando o subconjunto de teste. Ao final das iterações é calculada a acurácia sobre os erros encontrados.

Em virtude da quantidade de dados utilizados para o treinamento pode-se caracterizar este conjunto como um *dataset* simples, o que implica na facilidade de execução do SMO, uma vez que este não teve dificuldades em criar vetores de suporte para cada domínio. Portanto, todos os artigos foram classificados corretamente, atingindo um percentual de 100% de acerto. Obter esta precisão de acertos é considerado um evento raro no que tange à tarefa de classificação de textos. Contudo, esses valores podem ser explicados nesta situação pelo fato de que o *dataset* foi gerado de forma manual e com uma baixa quantidade de atributos e artigos inseridos. Diferentemente do sistema *UnderMine*, que lê os documentos textuais e a partir dele realiza todas as etapas.

6. Conclusão

A ferramenta *UnderMine Text Miner* utilizou uma nova implementação para a MT com base nos SIA, buscando agregar as melhores características de ambas as áreas científicas para a obtenção de um produto que execute a tarefa de MT de forma satisfatória e visando suprir as deficiências dos métodos atuais.

Durante os testes realizados foi possível identificar que o uso de SIA na tarefa de mineração produz resultados coerentes e robustos, demonstrando que o sistema é capaz de lidar com o diagnóstico dos mais diversos textos (antígenos). É válido afirmar, também, que agregar as principais características de ambas as áreas científicas demonstra que as metodologias imunológicas podem ser proveitosas nas mais diversas áreas computacionais. Futuramente, a inclusão de novos agentes e a utilização de bases diversificadas surgem como melhorias no sistema, uma vez que contribuem para o aumento da precisão de acerto no diagnóstico dos artigos através do enriquecimento de termos.

Referências Bibliográficas

ARANHA, C.N. **Uma abordagem de pré-processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional**, Tese de Doutorado, Departamento de Engenharia Elétrica, PUC-Rio, 2007.

DASGUPTA, D. **Artificial Immune Systems and Their Applications**, Springer-Verlag, 1999.

DE CASTRO, L.N. **Engenharia Imunológica**: Desenvolvimento e Aplicação de

Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais. Tese de Doutorado, UNICAMP, Faculdade de Eng Elétrica e Computação, 2001.

EBECKEN, N.F.; LOPES, M.C.S.; COSTA, M.C. A. Mineração de textos. In: REZENDE, S. de O. (Org.). Sistemas inteligentes. Barueri, SP: Manole, 2003. p. 337-370.

HOTHO, A.; NÜRBBERGER, A.; PAASS, G. A brief survey of text mining. Journal for Computational Linguistics and Language Technology, Vol. 20(1), pp. 19-62, 2005.

HOUJEIJ,A.; HAMIEH, L.; MEHDI, N.; HAJJ, H. A Novel Approach for Emotion Classification based on Fusion of Text and Speech, In: **Int Conf on Telecommunications**, p.1-6, 2012.

HSU, C.; CHANG, Y. Qualitative Text Mining in Student's Service Learning Diary. In: **Int Conf on Innovations in Bio-Inspired Computing and Applications** (IBICA), p.350-354, 2012.

LUPI, G.; PATELLI, P.; IACONESI, S.; PERSICO, O. DIY GIS: A Constructionist, Educational Toolkit for Architecture Students. In: **Int Conf on Advanced Learning Technologies (ICALT)**, p.253-257, 2012.

MA,J. ; XU, W. ; SUN,Y. ; TURBAN, E. ; WANG,S. ; LIU,O. An Ontology-Based Text-Mining Method to Cluster Proposals for Research Project Selection. **IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans**, v. 42 , Issue 3, p.784-790, 2012.

MORAIS, E. A.; AMBRÓSIO, A. P. L. **Mineração de Textos. Relatório Técnico.** Instituto de Informática; Universidade Federal de Goiás, 2007.

PENG,Y. ; TUDOR, C.O. ; TORII, M. ; WU, C.H. ; VIJAY-SHANKER, K. iSimp: A sentence simplification system for biomedicail text. In: **Int Conf on Bioinformatics and Biomedicine (BIBM)**, p.1-6, 2012.

PUSHPALATHA, K.P. ; RAJU, G. Compactness — A useful feature for generating search index, In: **Int Conf on Technology Enhanced Education (ICTEE)**, p.1- 6, 2012.

REATEGUI, E. ; KLEMANN, M. ; FINCO, M.D. Using a Text Mining Tool to Support Text Summarization, In: **IEEE 12th Int.Conf, on Advanced Learning Technologies (ICALT)**, p.607-609, 2012.

SACHIN, R.B.; VIJAY, M.S. A Survey and Future Vision of Data Mining in Educational Field. In: **Second Int.Conf, on Advanced Computing & Communication Technologies** (ACCT), p. 96-100, 2012.

WIVES, L. **Tecnologias de descoberta de conhecimento em textos aplicadas à inteligência competitiva.** Exame de Qualificação EQ-069, PPGC-UFRGS, 2002.

YANG, H.; LEE,C. Mining open source text documents for intelligence gathering. In: **Int Symposium on Information Technology in Medicine and Education** (ITME), p.969-973, 2012.

YI, Y.; Liu, L.; LI, C.; SONG, W. Machine Learning Algorithms with Co-occurrence based Term Association for Text Mining.In: **Int Conf on Computational Intelligence and Communication Network**., p.958-962, 2012.