RECIT
Revista Eletrônica Científica Inovação e Tecnologia
ISSN: 2175-1846

https://periodicos.utfpr.edu.br/recit

Mineração de textos na resolução de ambiguidades lexicais da Língua Portuguesa

Luiz Guilherme Fonseca Rosa
luizguilhermefr@gmail.com
Universidade Estadual do oeste do Paraná (UNIOESTE), Cascavel, Paraná, Brasil.

Jorge Bidarra
jorge.bidarra@unioeste.br
Universidade Estadual do oeste do Paraná (UNIOESTE), Cascavel, Paraná, Brasil.

RESUMO

Nos últimos tempos, a quantidade de textos que circula pela internet cresce a taxas surpreendentes, atingindo valores da ordem dos petabytes. As mídias sociais, portais de notícias e a própria literatura contribuem fortemente para este crescimento. Diante deste cenário, e tendo em vista que grande parte dos textos produzidos encontra-se armazenado em mídias eletrônicas, torna-se muito desejável transformá-los em conhecimento tangível e aplicável. Este artigo discute o fenômeno da ambiguidade lexical, problema recorrente no campo do Processamento de Linguagem Natural, apresentando alguns resultados obtidos ao aplicar técnicas de descoberta de Conhecimento em Bases de Dados e Aprendizagem de Máquina no tratamento do problema.

PALAVRAS-CHAVE: mineração de dados, processamento de linguagem natural, inteligência artificial.

INTRODUÇÃO

A quantidade de informação circulando no mundo cresce de um modo surpreendente. Estima-se que este volume dobra a cada 20 meses (Frawley et. al., 1992). Nossa capacidade para absorver e lidar com tanta informação vem nos exigindo, quer como pessoas, quer como instituições, muitos investimentos, notadamente em relação a sua captura e armazenamento (Chen et. al., 1996). Apenas para se ter uma ideia, em 2009, somente a rede social Twitter, com aproximadamente 41 milhões de usuários, registrava mais de 106 milhões de "tweets" – publicações de até 140 caracteres (Kwak et. al., 2010). O Wikipedia, portal de conhecimento colaborativo, possui hoje em suas bases de acesso mais de 5 milhões de artigos sobre os mais diversos temas e mais de 70.000 criadores de conteúdo (Wikipedia, 2016).

Armazenar, gerenciar e permitir a manipulação de grandes bases de dados, assegurando aos sistemas consistência, confiabilidade e rapidez, têm sido um dos grandes desafios para os especialistas da área de processamento automático da informação. Diversos trabalhos relacionados, não apenas ao Armazenamento e à Descoberta de Conhecimento em Bases de Dados (Larose et. al., 2014; Feldman et. al., 1995), mas, também voltados para Aprendizagem de Máquina (Mitchell, 1997; Witten et. al., 1999), vêm sendo desenvolvidos tanto por pesquisadores, quanto por profissionais da Ciência da Computação.

Neste artigo, são apresentados alguns resultados, ainda parciais, obtidos com pesquisas que estamos desenvolvendo no âmbito do Processamento da Linguagem Natural (PLN), mais diretamente relacionadas a análises e tratamento automático de ocorrências de palavras ambíguas encontradas em textos. Os módulos implementados vêm sendo usados por nossa equipe de trabalho como base de sustentação para a especificação, modelagem e implementação de um dicionário eletrônico em desenvolvimento.

Para tanto, o artigo assim se estrutura: Na Seção 1, são discutidos os aspectos teóricos tomados como base para a presente execução, assumindo-se como focos de debate o Processo de Descoberta de Conhecimento em Bases de Dados e a Mineração de Dados (KDD e MD) respectivamente, bem como Aprendizagem de Máquina (AM). Na Seção 2, aborda-se o problema da Ambiguidade Lexical, os dados envolvidos nos experimentos e a preparação dos dados para o processamento. Na Seção 3, passa-se à apresentação dos resultados, incluindo-se no debate análises correspondentes às estimativas de erro aferidas. Por fim (Seção 4), um modelo de classificação das sentenças com base nos sentidos assumidos pelas palavras ambíguas processadas é mostrado.

MÉTODOS

Define-se a Descoberta de Conhecimento em Bases de Dados ou Textos (Knowledge Discovery in Textual Databases – KDT) como sendo um processo não trivial, iterativo, interativo, composto por múltiplos estágios de processamento que vão desde a preparação dos dados à interpretação e validação dos resultados obtidos, passando pela extração de informações relevantes contidas nas bases analisadas, sendo esta, se não a mais importante etapa de todo o processamento, certamente a mais explorada pelos especialistas (Frawley et. al., 1992; Fayyad et. al., 1996). Etapa em que os dados são submetidos a algoritmos

de busca e identificação de padrões válidos, novos, potencialmente úteis e compreensíveis (Fayyad et. al., 1996), em via de regra, é nela que se concentra grande parte do processamento voltado para a descoberta de conhecimento. Combinada com técnicas de Aprendizagem de Máquina, a extração, ou Mineração de Dados, é capaz de promover a predição de dados futuros com base nos padrões encontrados (Witten et. al., 2011).

Embora seja a etapa central de todo o processo, a mineração deve ocorrer após a seleção, preparação e transformação, demais fases que o compõem. Partindo destes pressupostos, a ferramenta eleita para objeto de estudo deste artigo foi a Waikato Environment for Knowledge Analysis (WEKA) (Hall et. al., 2009), uma suíte contendo diferentes plataformas de trabalho para KDD, permitindo aos pesquisadores acesso fácil às diferentes técnicas de seu estado-da-arte. Dentre as utilidades da ferramenta, cita-se pré-processamento de dados, através de filtros e seleção de atributos, clustering, um método de aprendizado não-supervisionado, visualização gráfica de dados, e, por fim, a mineração propriamente dita, através de algoritmos de classificação, regressão e associação. Este compilado de utilidades faz do WEKA uma referência para experimentos na área, de modo em que diferentes conjuntos de dados e algoritmos de mineração podem ser facilmente avaliados de forma flexível. Além disso, através de um conjunto de rotinas em Java, a ferramenta permite acesso à sua interface através de uma Application Programming Interface (API), de modo que é possível incluir, de forma simples, chamadas aos seus métodos em qualquer sistema, sem ter de reimplementá-los um a um.

O problema da ambiguidade lexical face o processamento automático de textos

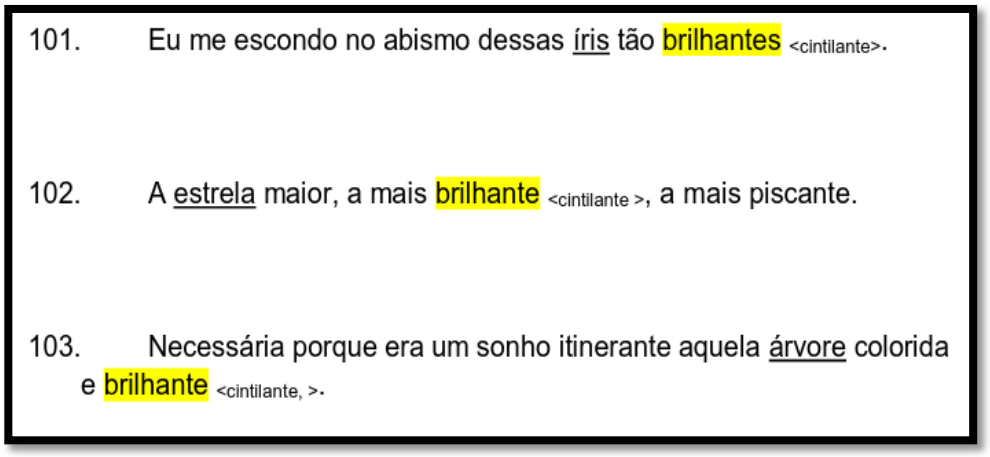
Línguas naturais são estruturadas em múltiplos níveis de forma simultânea. Isso inclui os níveis sintático, fonológico, morfológico, discursivo e léxico. Em qualquer ponto de determinada sentença, a informação expressa pode ser ambígua em um ou mais destes níveis (MacDonald et. al., 1994). A ambiguidade no nível lexical ocorre quando determinada palavra na sentença pode assumir mais de um significado. Exemplos de palavras ambíguas no nível léxico são banco, que pode assumir sentidos de entidade financeira e assento, brilhante, que pode assumir sentidos de notável e cintilante, e processo, que pode assumir sentidos de ação judicial ou procedimento. Este fenômeno representa um empecilho em diferentes áreas do Processamento de Linguagem Natural (PLN), ao passo em que o sistema computacional deve contornar todo tipo de ambiguidade.

Dado que a ambiguidade lexical é um problema para a linguística computacional, resulta que diferentes métodos de resolvê-la vêm sendo propostos pela academia, entretanto, a proposta deste trabalho se difere ao passo em que sugere o uso de técnicas de KDD em corpora, ou seja, bases textuais, a fim de aprender com estas bases a solucionar, parcial ou totalmente o problema. Parte-se do pressuposto de que um grande número de dados possui informação implícita suficiente para resolver estas e outras ambiguidades no âmbito de PLN.

Os dados selecionados para análise e experimento são frases variadas da língua portuguesa, contendo em si diferentes palavras ambíguas, organizadas em arquivos de acordo com a palavra estudada. Ao analisar-se cada um destes

arquivos (Figura 1), notaram-se alguns elementos que influenciavam, de alguma maneira, no sentido da palavra ambígua. Estes elementos foram escolhidos como atributos e formam o contexto semântico da palavra.

Figura 1 – Exemplo de dados para “brilhante”



Fonte: Autoria Própria (2016).

Na análise, também se incluíram, de forma manual, os significados de cada palavra, de tal maneira que posteriormente estes também representem um atributo, porém, este o atributo classe, ou seja, o atributo a ser predito após o aprendizado. Nota-se neste exemplo que o contexto semântico de “brilhante” para o sentido de “cintilante” é determinado pela existência de palavras como “íris”, “estrela” e “árvore”. Sabe-se, porém, que outras palavras podem determinar estes e outros contextos quando co-ocorrendo com brilhante, como astros, estrelas e joias. Ao total, foram selecionados em torno de 90 atributos. Para outras palavras, o procedimento ocorreu de maneira análoga.

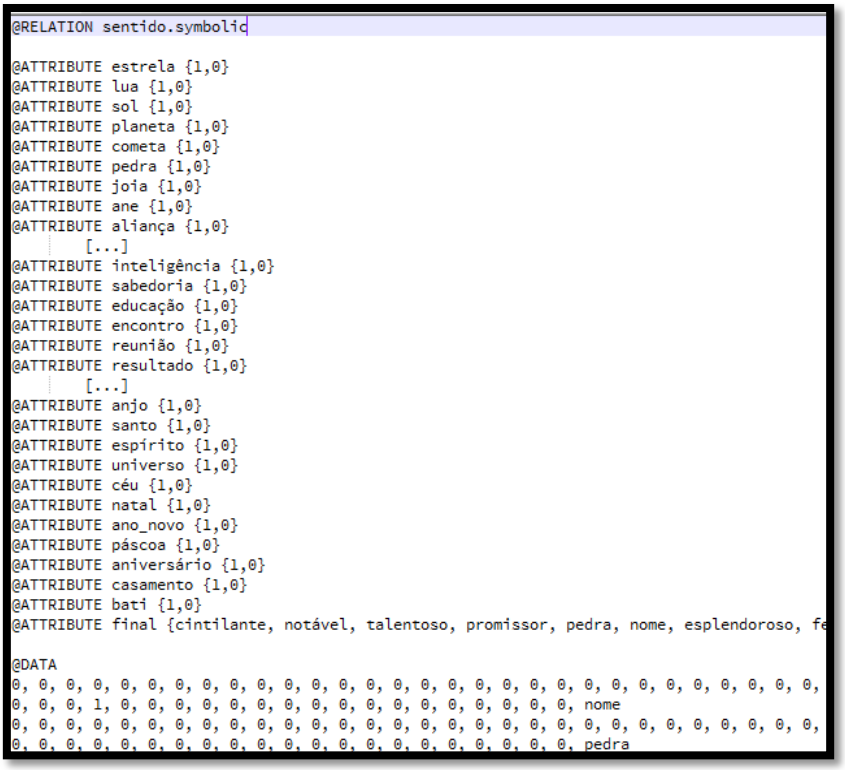
Ao finalizar a análise manual dos dados, a etapa seguinte é a transformação para o arquivo do tipo Attribute-Relation (ARFF), utilizado pelo WEKA. Basicamente, neste arquivo separam-se as variáveis (atributos) e os dados em função destas. Todos os atributos, exceto o atributo classe, foram definidos como tipo lógico (booleano), representando a existência ou não de cada palavra em cada sentença. As sentenças, por sua vez, foram representadas como conjuntos de “verdadeiro e falso” para cada atributo, finalizando com o atributo classe (Figura 2).

Apesar de ser uma etapa bastante intuitiva, deve-se tomar cuidado com a seleção de atributos. É esta fase que deve demandar maior tempo e atenção, pois trata-se da representação dos dados experimentais. Segundo Lee (2005), os atributos devem auxiliar na precisão e simplicidade do classificador. Além disso, atributos desnecessários e/ou redundantes afetam de maneira crucial a acurácia do modelo. Ainda segundo a autora, considera-se que atributos são redundantes entre si, parcial ou completamente, quando seus valores estão correlacionados. Portanto, a fim de eliminar redundância entre os atributos, eliminou-se do conjunto relações como "reunião" e "reuniões", "ladrão" e "ladrãozinho", etc.

Para tanto estes atributos foram lematizados, ou seja, substituídos pelo seu radical em comum, sem tempo verbal, plural ou gênero.

Com estes dados devidamente pré-processados e transformados, pode-se seguir para a fase de Mineração de Dados, onde, na ferramenta WEKA, estes serão submetidos a algoritmos de Aprendizado de Máquina e seus padrões explorados.

Figura 2 – Exemplo de arquivo tipo ARFF. Atributos são definidos com o prefixo @ATTRIBUTE e os dados em função destes são definidos abaixo de @DATA



```
@RELATION sentido.symbolic

@ATTRIBUTE estrela {1,0}
@ATTRIBUTE lua {1,0}
@ATTRIBUTE sol {1,0}
@ATTRIBUTE planeta {1,0}
@ATTRIBUTE cometa {1,0}
@ATTRIBUTE pedra {1,0}
@ATTRIBUTE joia {1,0}
@ATTRIBUTE ane {1,0}
@ATTRIBUTE aliança {1,0}
        [...]
@ATTRIBUTE inteligência {1,0}
@ATTRIBUTE sabedoria {1,0}
@ATTRIBUTE educação {1,0}
@ATTRIBUTE encontro {1,0}
@ATTRIBUTE reunião {1,0}
@ATTRIBUTE resultado {1,0}
        [...]
@ATTRIBUTE anjo {1,0}
@ATTRIBUTE santo {1,0}
@ATTRIBUTE espírito {1,0}
@ATTRIBUTE universo {1,0}
@ATTRIBUTE céu {1,0}
@ATTRIBUTE natal {1,0}
@ATTRIBUTE ano_novo {1,0}
@ATTRIBUTE páscoa {1,0}
@ATTRIBUTE aniversário {1,0}
@ATTRIBUTE casamento {1,0}
@ATTRIBUTE bati {1,0}
@ATTRIBUTE final {cintilante, notável, talentoso, promissor, pedra, nome, esplendoroso, fe

@DATA
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, nome
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, pedra
```

Fonte: Acervo próprio, 2016.

RESULTADOS E DISCUSSÃO

Nesta seção serão discutidos os resultados de dois algoritmos utilizados no processo de Mineração. Ambos encontram-se disponíveis para uso na ferramenta WEKA. São eles o J48, que é uma adaptação para Java do algoritmo de Quinlan "C4.5" (Quinlan, 1993) para árvores de decisão e o algoritmo SMO, que implementa uma adaptação do algoritmo de otimização sequencial mínima de John Platt's (Platt, 1998). Além disso, as estimativas de erro são baseadas na raiz quadrada da média dos quadrados dos erros unitários, ou Root Mean Squared Error (RMSE), medida amplamente difundida na literatura referente à MD. Sua fórmula é como:

$$\sqrt{\frac{(p_1-a_1)^2+\cdots+(p_n-a_n)^2}{n}} \qquad (01)$$

Onde p_1 à p_n são os valores preditos e a_1 à a_n os valores verdadeiros correspondentes.

O ambiente de validação adotado foi Cross-validation com 10 dobras. Este método consiste em decidir um número fixo de "folds" (divisões ou dobras entre os dados) e então, a cada turno, uma divisão é utilizada para validar e as demais para treinar. Isso se repete até que todas as instâncias do conjunto tenham sido utilizadas ao menos uma vez para validar. Testes extensivos em conjuntos de treinamentos numerosos e em diferentes técnicas de Aprendizado de Máquina já mostraram que 10 é o número ideal de dobras para se determinar a estimativa de erro mais acurada (Witten et. al., 2011). A máquina utilizada no experimento foi uma Intel Core i3-2100 de 3.10 Ghz, 6 GB de memória RAM e Sistema Operacional Linux Ubuntu 15.10 64bits.

SMO: O algoritmo SMO é encontrado no WEKA sob a categoria "funções". Encontramos, neste grupo, classificadores que podem ser vistos de uma forma direta como equações matemáticas. Os resultados obtidos com este algoritmo são bastante otimistas, todavia seu custo computacional e taxa de erro se sobressai ao seu concorrente neste experimento. O resultado é descrito na Tabela 1.

Tabela 1 - Resultados obtidos com SMO

Conjunto	Acerto (%)	Erro (%)	RMSE	Tempo (s)
Banco	94,06	5,94	0,3118	0,17
Brilhante	65,66	34,34	0,3039	≈ 0.1

Fonte: Autoria Própria (2016).

J48: O algoritmo J48 é uma reimplementação do C4.5, algoritmo já consagrado na área de MD. Este utiliza de Árvores de Decisão para construir um modelo de predição, selecionando um atributo para encabeçar o topo (raiz), e então dividindo este nó para cada valor possível a partir dele, criando novos ramos com novos nós, e a partir destes nós, novas divisões de forma recursiva, no processo de "dividir e conquistar". O resultado é descrito na Tabela 2.

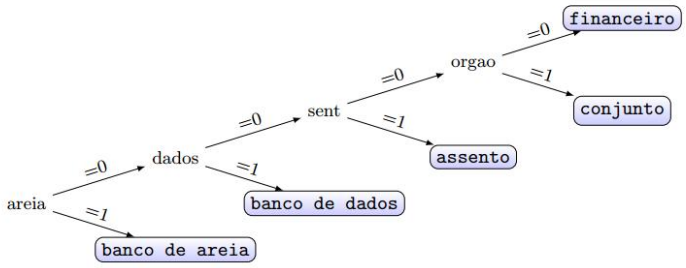
Tabela 2 - Resultados obtidos com J48

Conjunto	Acerto (%)	Erro (%)	RMSE	Tempo (s)
Banco	94,52	5,48	0,1317	0,05
Brilhante	61,62	38,38	0,2556	≈ 0.0

Fonte: Autoria Própria (2016).

Conclusões: Embora o algoritmo SMO tenha se mostrado um concorrente a altura, a eficiência e baixo custo computacional do algoritmo J48 fazem deste a escolha favorita ao se tratar do conjunto de dados em questão. Como estamos tratando de dados binários, ou seja, existência ou não existência de atributos em cada sentença, uma árvore de decisão mostra-se a escolha mais adequada de modo que pode-se, partindo de sua raiz, seguir caminhos bem definidos até chegar em alguma folha. Veja na Figura 3 uma análise gráfica dessa situação.

Figura 3 – Árvore de Decisão reduzida para "banco"



Fonte: Acervo próprio, 2016.

Aplicação Prática do Modelo Extraído

Embora muitas das vezes a intenção do pesquisador com o processo de KDD seja mera extração visual e estatística para análise humana, o problema que se discute aqui exige um aprofundamento computacional a mais. Os modelos e padrões aqui extraídos não demonstram grande descoberta além do que já era intuitivo: determinadas palavras influenciam no sentido de outras. A questão que aqui se quer chegar é ensinar isso ao computador através do Aprendizado de Máquina e utilizar tal conhecimento em aplicações reais do âmbito da Linguística Computacional, seja para Tradução Automática, Interação Humano-Computador ou outras tantas aplicações citáveis neste campo. Todavia, para esta análise, deve-se empregar o modelo extraído em um sistema computacional desenvolvido sob demanda para este problema.

O sistema aqui citado é um programa desambiguador, útil, por exemplo, para semiautomatização do processo de anotação da Seção 2, ou para emprego em um sistema maior que o adote como módulo. O sistema foi desenvolvido em Java, empregando a API do WEKA de maneira a chamar o J48 e outros métodos sem ter de reescrevê-los.

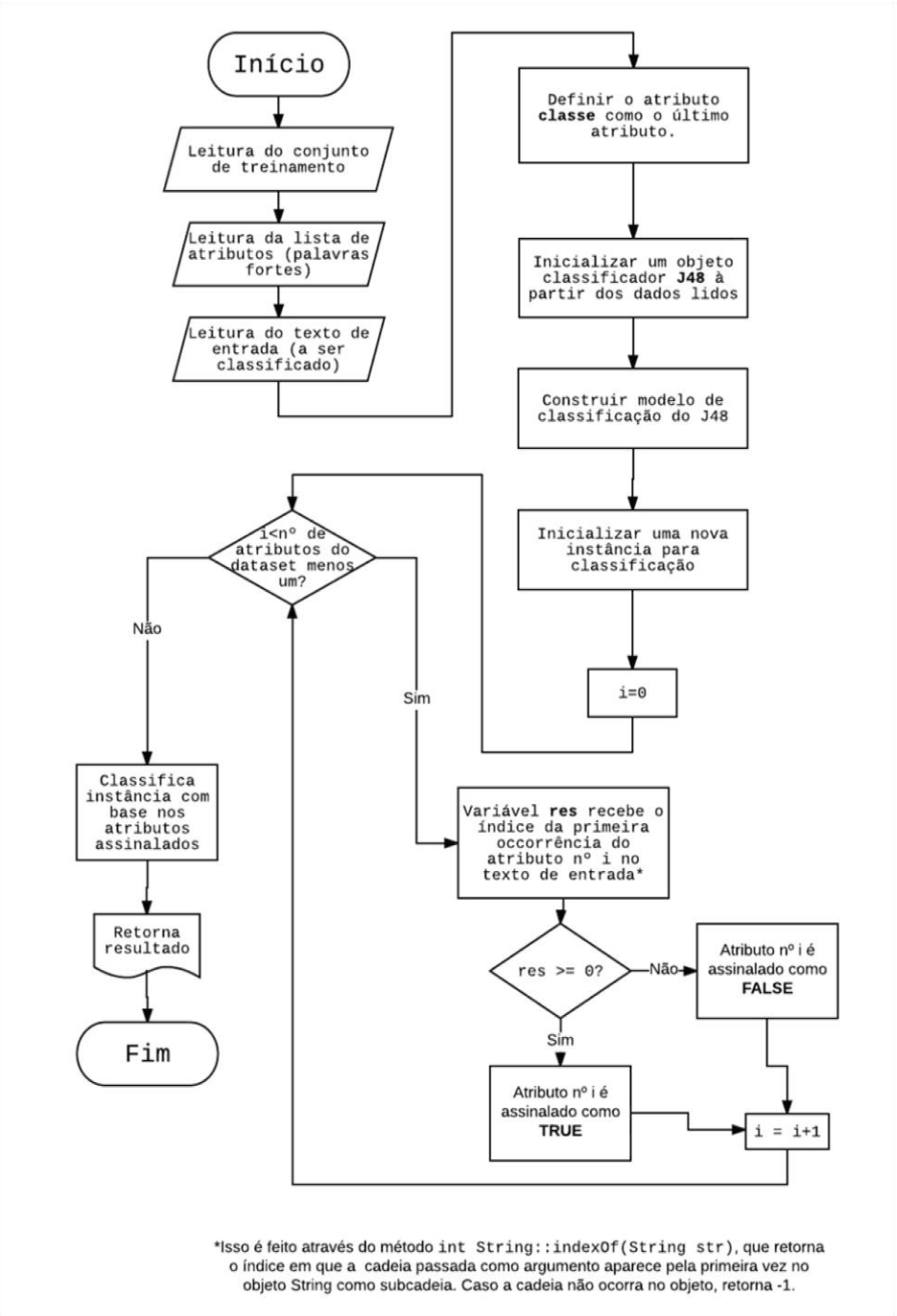
A classificação em uma interface como esta ocorre da seguinte maneira: Primeiramente, inclui-se a biblioteca referente ao WEKA através de um arquivo no diretório do projeto. Em seguida, faz-se um chamamento do arquivo ARFF, o mesmo utilizado nos experimentos com a ferramenta, já através de um método da mesma. Executa-se um classificador J48, também pela API, carregando o modelo na memória e então lê-se a entrada e classifica-a conforme este modelo. Este formato pode ser melhor visualizado no diagrama da Figura 4.

Ao somar o poder de classificação do WEKA à métodos Java para manipulação de arquivos, strings e outros elementos, pode-se formar uma ferramenta verdadeiramente útil a diferentes problemas da era da informação. O programa por nós desenvolvido envolve estas utilidades, além de uma interface gráfica que possibilita à qualquer interessado efetuar testes e validações no modelo estabelecido.

Todavia, este programa é apenas um modelo, que pode e deve ser aprimorado e adaptado a diferentes realidades. Sua implementação em código aberto está disponível através do endereço <https://github.com/luizguilhermefr/porlibras>.

Mostra-se, com isso, que o processo de Descoberta de Conhecimento em Bases de Dados é útil também para o processamento de línguas naturais, tendo o pesquisador de ter a afinidade necessária com o problema e o conhecimento dos dados que se tem em mãos.

Figura 4 – Fluxograma do núcleo de classificação implementado



Fonte: Autoria Própria (2016).

Text mining in the Portuguese lexical ambiguity resolution

ABSTRACT

In the last years, the amount of texts circulating the internet has growth dramatically, reaching the order of petabytes. The social media, news portals and even the literature contribute strongly for this increase. In front of this scenario, and having in mind that the majority of the produced texts are stored in electronic media, becomes very desirable to transform it in applicable and tangible knowledge. Thus, this paper discusses the lexical ambiguity phenomenon, recurrent problem in the Natural Language Processing Field, presenting some results obtained by applying Knowledge Discovery in Databases and Machine Learning techniques in the problem treatment.

KEYWORDS: data mining, natural language processing, artificial intelligence.

NOTAS

1 IMAGEPARSING. Disponível em:

http://www.imageparsing.com/Dataset_Album/. Acesso em: 30/07/2015.

REFERÊNCIAS

CHEN, M.; HAN, J.; YU, P. S. **Data Mining: An Overview from a Database Perspective**. IEEE Transactions On Knowledge and Data Engineering Vol. 8 No. 6, p. 866-883, 1996.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery in Database.** In: AI Magazine, Volume 17 Number 3, p. 37-54, 1996.

FELDMAN, R.; DAGAN, I. **Knowledge Discovery in Textual Databases (KDT)**. AAAI KDD-95 Proceedings, p. 112-117, 1995.

FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. **Knowledge Discovery in Databases: An Overview**. AI Magazine Volume 13 Number 3, p. 57-70, 1992.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. **The WEKA Data Mining Software: An Update**. SIGKDD Explorations, Volume 11, Issue 1, p. 10-18, 2009.

KWAK, K.; LEE, C.; PARK, H.; MOON, S. **What is Twitter, a Social Network or a News Media?** Proceedings of the 19th International World Wide Web (WWW) Conference, p. 591-600, 2010.

LAROSE, D. T.; LAROSE, C. D. **Discovering Knowledge in Data: An Introduction to Data Mining, 2nd Edition.** John Wiley & Sons, 2014.

LEE, H. D. **Seleção de atributos importantes para a extração de conhecimento de bases de dados**. USP São Carlos, 2005.

MACDONALD, M. C.; PEARLMUTTER, N. J.; SEIDENBERG, M. S. **Lexical Nature of Syntactic Ambiguity Resolution**. Psychological Review, Vol. 101 No. 4, p. 676-703, 1994.

MITCHELL, T. **Machine Learning**. McGraw Hill, 1997.

PLATT, J. **Advances in Kernel Methods - Support Vector Learning**. MIT Press, 1998.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. Morgan Kaufmann Publishers, 1993.

Wikipedia: About. Disponível em: <https://en.wikipedia.org/wiki/Wikipedia:About>, Acesso em: 12 ago 2016.

WITTEN, I. H.; FRANK, E. **Data Mining - Practical Machine Learning Tools and Techniques With Java Implementations**. Morgan Kaufmann Publishers, 1999.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining - Practical Machine Learning Tools and Techniques 3ª Ed.** Morgan Kaufmann Publishers, 2011.

Recebido: 16 ago. 2016.
Aprovado: 23 nov. 2016.
DOI:
Como citar: ROSA, L. G. F.; BIDARRA, J. Mineração de textos na resolução de ambiguidades lexicais da Língua Portuguesa. R. Eletr. Cient. Inov. Tecnol., Medianeira, v. 2, n. 14, p. 127-138, jul./dez. 2016. Disponível em: <https://periodicos.utfpr.edu.br/recit>. Acesso em: XXX.
Correspondência:
Luiz Guilherme Fonseca Rosa
Rua Lions Clube, 710, Jardim Maria Luiza, Cascavel, Paraná, Brasil.