

Atividade Avaliativa II - Sistemas Especialistas

Giovanni F. S. Rebouças¹, Felipe A. da S. Biazatti², Pedro A. B. Bispo³

^{1,2,3}Institute of Technological Sciences, Universidade Federal de Itajubá, Itabira, Minas Gerais, Brazil

RA: 34270¹, 34452², 2016002449³

giovannireboucas@hotmail.com, felipeamorimsbiazatti@gmail.com, pedro_augustobastos@hotmail.com

Resumo—Esse trabalho visa fazer uma breve explicação sobre o que são Sistemas Especialistas (SE), quais são suas possíveis aplicações e, então, exposto esse conhecimento, implementar um SE que haja como ferramenta investigativa e diagnóstica para acidentes ofídicos. Esse SE foi implementado através, inicialmente, do Expert SINTA e então re-implementado baseando-se em um Shell implementado em Python.

Esse sistema especialista visa oferecer uma ferramenta investigativa e diagnóstica para profissionais da área de medicina e cidadãos que podem se deparar com acidentes ofídicos no Brasil. O sistema, através dos sinais e sintomas, é capaz de diagnosticar o agente causador do acidente e a gravidade de atuação da peçonha e então, com isto, ser capaz de sugerir um tratamento por soroterapia adequado.

Palavras-chave—Sistemas Especialistas, Acidentes Ofídicos, Python, Expert SINTA, Soroterapia.

I. INTRODUÇÃO

Com o advento da terceira revolução industrial, chamada também de Revolução Informacional, que começou em meados do século XX, nós buscamos maneiras inovadoras e disruptivas para resolver os problemas que nos cercam.

Em 19 de abril de 1965 *Gordon E. Moore* publicou um artigo de cunho científico pela revista *Electronic Magazine* e fez a seguinte citação: "A complexidade para componentes com custos mínimos tem aumentado em uma taxa de aproximadamente um fator de dois por ano ... Certamente em um curto prazo pode-se esperar que esta taxa se mantenha, senão aumentar. A longo prazo, a taxa de aumento é um pouco mais incerta, embora não haja razões para se acreditar que ela não se manterá quase constante por pelo menos 10 anos. Isso significa que em torno de 1975, o número de componentes por circuito integrado para um custo mínimo será 65 000 (65nM). Eu acredito que circuitos grandes como este poderão ser construídos em um único componente." Isso ficou conhecido posteriormente como **Lei de Moore**, e dado essa evolução em poder computacional ao passar dos anos a quantidade de problemas a serem resolvidos por computadores aumentaram drasticamente.

Computadores agora são capazes de realizar tarefas de alta complexidade e com extrema precisão permitindo que eles substituíssem atores humanos em tarefas com alto grau de periculosidade, que requerem alta precisão ou cujos resultados impactam seriamente a vida de si ou de outros. Isso permitiu que áreas novas e tarefas não pensadas antes surgissem, como o campo de inteligência artificial e sistemas especialistas que serão os temas abordados neste artigo.

II. REVISÃO DE LITERATURA

A principal fonte consultada sobre conhecimento com relação à acidentes envolvendo animais peçonhentos foi o Manual de Diretrizes Diagnósticas de Acidentes com Animais Peçonhentos da Universidade Federal do Rio de Janeiro (UFRJ). [1] Nesse, há discriminado informações diagnósticas não apenas para casos de ofidismo, mas para casos que envolvem outros animais como, escorpionismo, araneísmo, himenopterismo e outros.

Todas as variáveis, variáveis objetivos e regras para implementação do SErpente foram tirados diretamente do capítulo de ofidismo desta publicação sendo a principal e única fonte de informação para isto para que não houvessem divergências com relação à maneira que outros materiais buscavam diagnosticar e tratar acidentes desta natureza.

A. Ofidismo

O capítulo de ofidismo do Manual de Diretrizes Diagnósticas foca na identificação, detalhamento de características físicas e da peçonha, manifestações clínicas e a soroterapia para os 4 gêneros de cobras mais comuns no Brasil e que compõe quase 80% dos casos de ofidismo em território nacional. Os acidentes e os respectivos gêneros dos agentes causadores são:

- Botrópico (gênero *Bothrops*) - Compõe 70% dos acidentes ofídicos no Brasil e cujo agente causador é popularmente conhecida como Jararaca;
- Crotálico (gênero *Crotalus*) - Compõe 6% dos acidentes ofídicos no Brasil e cujo agente causador é popularmente conhecida como Cascavel;
- Laquétrico (gênero *Lachesis*) - Compõe 1% dos acidentes ofídicos no Brasil e cujo agente causador é popularmente conhecida como Surucucu;
- Elapídico (gênero *Micrurus*) - Compõe um pouco menos de 1% dos acidentes ofídicos no Brasil e cujo agente causador é popularmente conhecida como Coral Verdadeira;

O capítulo de ofidismo ainda provém informações suficientes sobre os sinais clínicos e sintomas apresentados pelos pacientes para que possa-se fazer uma estimativa grosseira sobre quanto tempo faz desde que o acidente ocorreu e sobre a gravidade do caso, podendo ser leve, moderado e grave.

Munido das informações sobre o agente causador, tempo desde o acontecimento do acidente e a gravidade do caso, o manual ainda provém sugestões de tratamento por soroterapia adequados, especificando tanto o princípio ativo quanto a dosimetria medida em número de ampolas padrões (10mL) [2]

III. MATERIAIS E MÉTODOS

Como pré-projeto, optou-se por inicialmente implementar o SERpente no Expert SINTA para que, depois de averiguado o funcionamento do sistema especialista que inclui o ajuste grosso e fino das regras, variáveis e variáveis objetivo, pudéssemos re-implementar o sistema em algum outro shell. No caso desse trabalho, optamos por fazer a implementação com base em um shell barebones desenvolvido em Python pelo Viktor Berger [3]. Detalhes de implementação serão discutidos na seção de Resultados.

A. Expert SINTA

O Expert SINTA é uma ferramenta computacional em que são utilizadas técnicas de inteligência artificial na sua aplicação, para geração automática de sistemas especialistas. Esta ferramenta baseia-se em conceitos utilizados em estatística, probabilidades e regras de produção para simplificar o trabalho de implementação destes sistemas. Em sua arquitetura de software, o Expert Sinta utiliza um motor de inferência compartilhada, com construção de telas e menus de forma automática, realiza o tratamento de probabilidades das regras de produção e utiliza-se de explicações sensíveis ao contexto da base de conhecimento modelada. Este software é de grande utilidade em problemas de classificação como o abordado neste trabalho. [4]

Após construção e implementação do sistema especialista de interesse, sua interface consiste na resposta de uma sequência de menus que podem se encaixar de forma bastante flexível a variados temas e objetivos propostos que gerarão ao final da pesquisa, resultados de acordo com o que foi construído em sua implementação, estes podem variar desde diagnósticos médicos, configuração de redes de computadores e no trabalho atual, a sugestão de soroterapia que irá ser aplicada a um paciente que sofreu acidente ofídico. [4]

B. Python

A linguagem Python nasceu no começo da década de 90, criado pelo matemático e programador Guido Van Rossum. Ele a projetou para dar ênfase ao trabalho do desenvolvedor, simplificando a escrita de um código limpo, simples e legível. Apesar de ser uma linguagem de alto nível, é utilizada nas mais diversas aplicações, web, servidores e ciência de dados.

Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. A linguagem segue um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation.

IV. RESULTADOS

Abaixo expusemos um resumo bastante grosseiro sobre os detalhes das implementações do SERpente no ambiente do Exper SINTA e do shell proprietário escrito e re-adequado em Python.

A. Serpente no Expert SINTA

A implementação do SERpente no Expert SINTA se deu com o objetivo primordial de funcionar como um ambiente de pré-projeto onde a definição das regras, variáveis e variáveis objetivos pudessem ser desenvolvidos com maior facilidade e transparência sobre os aspectos operacionais do Sistema Especialista.

Com base nas informações extraídas do capítulo de ofidismo do Manual de Diretrizes Diagnósticas da UFRJ e com enfoque nas informações expostas na seção de Revisão de Literatura, pudéssemos estabelecer um conjunto de 26 regras, que fazem uso de e manipulam 32 variáveis diferentes com o objetivo de atingir valores para 3 variáveis objetivos, configurando um conjunto de 8 possíveis combinações de resultados.

As variáveis objetivos do SERpente são:

- "gênero da serpente" - Identifica o gênero da serpente que causou o acidente ofídico e pode assumir os valores de: botrópico, laquético, elapídico ou crotálico;
- "gravidade" - Identifica a gravidade do acidente ofídico e pode assumir os valores de: leve, moderado ou grave; e
- "soroterapia" - Identifica qual soro e qual a dosimetria adequada deste e pode assumir os valores de: "SAB/SABC/SABL, 2-4 ampolas", "SAB/SABC/SABL, 4-8 ampolas", "SAB/SABC/SABL, 12 ampolas", "SAC/SABC, 5 ampolas", "SAC/SABC, 10 ampolas", "SAC/SABC, 20 ampolas", "SAL/SABL, 10-20 ampolas" ou "SAE, 10 ampolas".

Para conseguir associar valores às variáveis objetivos especificada acima, as regras do SERpente foram desenvolvidas e são separadas nos seguintes blocos lógicos:

- Definição da suspeita do agente causador - Conjunto de 4 regras cujos objetivos são definir qual é a suspeita do gênero da serpente que causou o acidente a partir dos sinais clínicos apresentados pelo paciente;
- Confirmação da suspeita do agente causador - Conjunto de 6 regras cujos objetivos são confirmar alguma suspeita de gênero da serpente, fazendo uso de sinais que podem ter aparecidos por um atendimento tardio do paciente. A partir desta regra, a variável objetivo do "gênero da serpente" já é definido;
- Definição da gravidade do caso - Conjunto de 8 regras cujos objetivos são definir qual é a gravidade do caso com base no gênero da serpente e da presença ou ausência de sinais clínicos que podem indicar a gravidade. A partir desta regra, a variável objetivo da "gravidade" já é definida; e
- Definição da soroterapia - Conjunto de 8 regras cujos objetivos são definir qual é o soro e dosimetria adequadas a serem administrados para tratar o paciente, fazendo uso do gênero da serpente e da gravidade do caso. A partir desta regra, a variável objetivo da "soroterapia" já é definida.

Note que as regras especificadas acima respeitam a ordem especificada no Expert SINTA e note que é fora do escopo deste trabalho especificar todas as regras detalhadamente uma vez que estas podem ser analisadas nos arquivos de implementação do

sistema especialista. Os arquivos de implementação do sistema no Expert SINTA se resumem ao `SERpente.bcm` e deve ser aberto através do uso do Expert SINTA.

B. *SERpente em Python*

A re-implementação do *SERpente* em um shell alternativo que não fosse o Expert SINTA possuiu como objetivos a familiarização com uma maneira alternativa de implementação de sistemas especialistas que tivesse uma aceitação internacional e acadêmica maior ou que possuísse uma maior facilidade de integração, além de prover um entendimento mais profundo por trás do funcionamento adequado de um sistema especialista ao se possuir um conjunto de regras, variáveis e variáveis objetivos cujos resultados fossem previsíveis e, desta maneira, pudessem ser usados para diagnosticar o funcionamento adequado ou não da re-implementação.

Como mencionado anteriormente, a implementação em Python faz uso de um shell desenvolvido por um autor croata chamado Viktor Berger em 2014 [3]. Esta implementação havia sido feito em 3 arquivos separados que compunham a execução do sistema especialista em um, o *parse* do arquivo de base de conhecimento em outro e o arquivo de base de conhecimento propriamente dito no terceiro arquivo (em formato `.txt`). Esta implementação havia sido feito totalmente em **croata** e feita em **Python 2** e, por conta disto, precisou ser traduzida e reimplementada não apenas para aspectos de entendimento do código mas para garantir que o funcionamento do mesmo seria preservado.

Os 2 arquivos Python que compunham a camada operacional do shell foram traduzidos para inglês e fundidos em 1 só arquivo para que todas as funções estivessem disponíveis de maneira monolítica, dispensando o uso de `import s`. Após a tradução da camada operacional e a fusão dos 2 arquivos, foi feito a tradução de um dos arquivos de exemplo que existiam no repositório original do autor para que pudesse ser usado como base para entendimento de funcionamento e *debugging* da implementação. Com isto, o código foi todo refatorado para se livrar das implementações de funções nativas do Python 2 em favor do Python 3.

Após todo o processo de tradução e refatoração do código para que ele funcionasse plenamente em Python 3, tomou-se a decisão para re-implementar as funções existentes usando

Todo este processo se tomou utilizando o arquivo de teste originalmente ofertado pelo autor original e traduzido do croata para o inglês. Após atestar o pleno funcionamento do Shell, iniciou-se o processo de re-implementar o *SERpente* no formato de base de conhecimento com o qual o Shell estava acostumado à tratar, com ênfase principal em garantir que o sistema funcionasse tão bem quanto funcionava no Expert SINTA, mantendo a ordenação lógica das regras e fazendo o mínimo de alterações necessárias para que a etapa de *parse* e subsequentes

Orientação à Objeto para que se pudesse tirar proveito das qualidades advindas deste tipo de implementação em Python. Foi feito também um trabalho de comentar todas as funções e etapas do código para garantir facilidade no entendimento do funcionamento do shell.

derivações inerentes da operação do shell recebesse regras e variáveis nos formatos esperados.

Por conta da maneira como *parse* do arquivo de base de conhecimento é feito, exista uma diferença primordial que ocorre entre as implementações do sistema no Expert SINTA e em Python, que é a existência de um aumento quase imperceptível de regras (de 26 para 33) para se adequar à maneira como o shell em Python trata o arquivo de base de conhecimento.

Note que, como citado na seção anterior, é fora do escopo deste trabalho especificar todas as regras detalhadamente uma vez que estas podem ser analisadas nos arquivos de implementação do sistema especialista. Os arquivos de implementação do sistema no shell proprietário em Python se resumem à implementação do shell descrito em `SERpente.py` e à descrição das variáveis, regras e objetivos em `SERpente.txt`. A execução e teste do Shell deve ser feita ao se invocar o comando `python SERpente.py` em uma máquina que possua uma instalação válida de **Python 3**.

V. CONCLUSÃO

Com a crescente interdisciplinaridade dos problemas que a computação passa à ser envolvida, com o inerente crescimento, popularização e acessibilidade de sistemas informatizados, e com o crescente interesse em resolver problemas não-computacionais, com sistemas, técnicas e metodologias computacionais são o racional por trás do desenvolvimento de sistemas especialistas.

Os sistemas especialistas são a evolução informatizada e constantemente acessível dos especialistas humanos em tópicos determinísticos ou probabilísticos que podem ser respondidos ou resolvidos através de um conjunto de conhecimento e regras de inferência. Problemas relacionadas com diagnósticos, tomadas de decisões e informações catalogáveis são todos problemas passíveis de resolução através do uso de sistemas especialistas.

REFERÊNCIAS

- [1] Alberto Chebabo Livia Batista Leis. Diretrizes diagnósticas de acidentes com animais peçonhentos. <http://www.hucff.ufrrj.br/download-de-arquivos/category/26-dip?download=332:rotinas>, 2018.
- [2] Maurício Ando. Bula do soro antibotrópico (pentavalente) e antilaquético. [https://butantan.gov.br/assets/pdf/soros_vacinas/soros/frmVisualizarBula%20\(1\).pdf](https://butantan.gov.br/assets/pdf/soros_vacinas/soros/frmVisualizarBula%20(1).pdf), 2016.
- [3] Viktor Berger. Simple expert-system written in python. <https://github.com/ViktorBerger/Expert-System>, 2014.
- [4] Grupo SINTA. Laboratório de Inteligência Artificial (LIA). Expert sinta - manual do usuário. <https://www.cin.ufpe.br/~fab/expert-sinta/manual.pdf>, 1995.