Revisão sobre interfaces de processamento de linguagem natural

Felipe A. Barusso[1], Danilo Yudi Futata Kassuya[2], Guilherme Henrique Gonçalves Silva[3]

[1]Departamento de Computação – Universidade Estadual de Londrina (UEL) Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

felipe.barusso@uel.br,danilo.yudi.futata@uel.br,

guilherme.henrique.silva@uel.br

Abstract. *The objective of this article is to gather information about many types of programs that use natural language to create an interface that is easy to understand and has high interactivity.*

Resumo. *O objetivo deste artigo é reunir informação sobre diversos programas que usam da linguagem natural para criar uma interface de fácil entendimento e de alta interatividade.*

1. BASEBALL

BASEBALL é um exemplo pioneiro de sistemas de tradução automática, entendimento e geração de fala, respondendo perguntas e sumarização se originaram na área da Inteligência Artificial e do Processamento de Linguagem Natural. Os primeiros programas em linguagem natural procuravam obter somente resultados limitados em domínios específicos. BASEBALL que funcionavam como interface em linguagem natural entre o usuário e um banco de dados [1].

BASEBALL usou um arquivo de dados contendo o mês, dia, lugar, equipes e resultado de cada jogo de baseball disputado durante uma época do campeonato americano. Respondia quase que todas as perguntas com os dados armazenados. Veja como seriam as perguntas traduzidas. Exemplos: “Para quem perdeu a equipe Red Sox em 19 de junho?” “Qual foi o resultado do jogo em Nova Iorque em 26 de setembro?”[2].

2. SAD-SAM

SAD-SAM foi um programa escrito por Robert K. Lindsay(1960).O programa entendia sentenças curtas e as utilizava para tirar algumas conclusões lógicas simples relacionadas a parentescos familiares.

O sistema utilizaria as regras de estrutura frasal propostas por Chomsky(1957) realizando uma análise de uma frase curta que resultaria em um gráfico. Após construir o gráfico da sentença o programa realizaria uma análise semântica dos substantivos da frase tentando relacionar ao que ou quem eles se referem marcando essas instâncias de substantivos como equivalentes, em seguida o programa procura por palavras que representem relações familiares como "father," "mother," "brother," "sister," "offspring," "brother-in-law," "sister-in-law" e"married.", com isso o programa desenvolve uma árvore familiar, e por fim o programa fará algumas conclusão para informações que não foram definitivamente confirmadas ou que são improváveis,

preenchendo algumas listas com as possibilidades para cada relacionamento ou com as impossibilidades[3].

They are flying planes swiftly

1. NP + V + V + N + Ad
   can go no farther
2. NP + V + V + NP + Ad
   can go no farther
3. NP + V + A + N + Ad
   NP + V + NP + Ad
   S + Ad
   can go no farther
4. NP + V + A + NP + Ad
   can go no farther
5. NP + Aux + A + N + Ad
   NP + Aux + NP + Ad
   can go no farther
6. NP + Aux + A + NP + Ad
   can go no farther
7. NP + Aux + V + N + Ad
   NP + VP + N + Ad
   can go no farther
8. NP + Aux + V + NP + Ad
   NP + VP + NP + Ad
   - 8.1. S + Ad
     can go no farther
   - 8.2. NP + Pred
     S
     successful parsing

Figura 1. Exemplo de uma análise

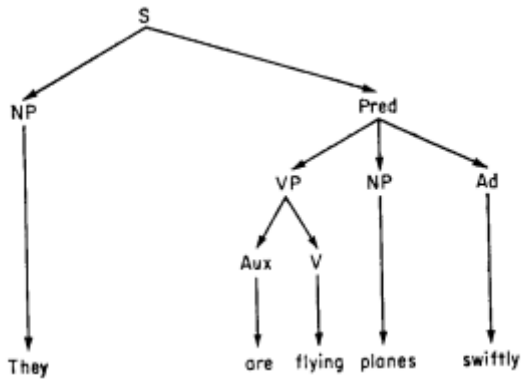

Figura 2. Gráfico resultante da análise

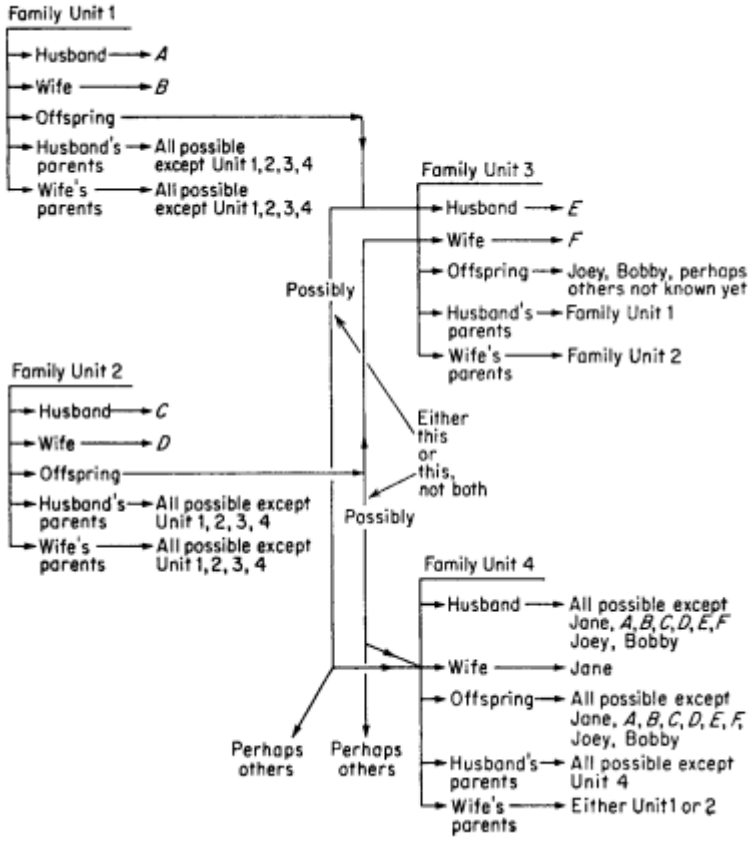

Figura 3. Exemplo de árvore familiar final

3. EIIZA

Criado em 1966, foi o mais citado de todos os programas em LN, desenvolvido por Joseph Weizenbaum, está escrito de forma a assumir o papel de um psiquiatra em conversação com um paciente humano.

As sentenças de entrada eram simples interrogativas, do qual o programa retirava palavras-chaves que eram utilizadas para encontrar a resposta. Por causa do domínio restrito e do processamento de palavras-chaves, estes sistemas ignoram a complexidade da linguagem.

Os princípios usados pelo ELIZA são muito fáceis de descrever. Inicialmente, varre a entrada em busca de palavras-chave, que, quando detectadas, disparam sempre a mesma ação, que consiste em ou devolver uma mensagem padrão, ou usar parte da entrada para construir uma mensagem. Por exemplo, se o usuário digitasse:

Odeio sorvete

O programa detectaria a palavra-chave “odeio” e responderia.

Não é bom odiar

Observe que esta resposta era dada, independentemente do restante da frase. Se isso fosse tudo o que ELIZA fizesse, ficaria muito fácil descobrir que se tratava de um programa, pela quantidade limitada de respostas que daria. Entretanto, para introduzir alguma variação em suas respostas, o programa fazia uso de sentença de entrada.

Qualquer sentença que o usuário digitasse seria varrida em busca de certas palavras ou frases tais como “meu” é transformado em “seu” ou “você é” em “eu sou”. A finalidade destas transformações simples é devolver ao usuário as sentenças que tenha introduzido, como se estas fossem geradas pelo programa. Por exemplo, recebendo a sentença:

Você é um idiota

o computador devolveria

sou um idiota

possivelmente acompanhada de alguns sinais de exclamação ou de interrogação. Estas duas técnicas, resposta a palavras-chaves e alterações de tempos verbais, acompanhada de alguns outros truques especializados, podem produzir um programa que mantenha uma conversação razoável com o usuário [4].

4. STUDENT

O STUDENT é uma aplicação de inteligência artificial que foca em resolver problemas de álgebra descritos em linguagem natural. O programa foi desenvolvido em LISP por Daniel G. Bobrow como sua tese de doutorado em 1964.

Segundo o autor, o programa aceita “um subconjunto confortável, mas restrito de inglês” [5] utilizado para expressar problemas algébricos. O mesmo faz uso de um repositório de informações que não pertencem a problemas específicos.

Para realizar a resolução dos problemas, o STUDENT faz suposições sobre a interpretação de ambiguidades na formulação do enunciado do problema. Caso tal suposição ocorra, o usuário é informado.

Na prática, o programa processa a entrada e a transforma em equações. Em seguida, tenta resolver o problema e encontrar o valor das variáveis. Caso um valor seja encontrado, a resposta é informada ao usuário, caso contrário, a aplicação solicita ao usuário mais informações.

Exemplo de cadeia de entrada: “If the number of customers Tom gets is twice the square of 20% of the number of advertisements he runs, and the number of advertisements is 45, then what is the number of customers Tom gets?” [6].

5. PROTO-SYNTHEX I

O Proto-synthex é um programa de processamento de linguagem natural desenvolvido para o IBM 7090 que recebe como entrada perguntas (em inglês) e tenta devolver ao usuário uma resposta. Seu nome significa a síntese de material verbal complexo. A aplicação opera utilizando o conceito de *natural language store information* [7], armazenando dados de entrada.

O armazenamento de informações é realizado na forma de frases em inglês comum retiradas do que foi digitado pelo usuário, essencialmente sem edição. Em suma, tudo que o programa recebe como entrada é armazenado em uma fita magnética.

6. SIR

O SIR (ou NLP-SIR) é uma interface de linguagem natural para aquisição de informações de planilhas. O sistema permite aos usuários realizar tarefas comuns como filtrar e gerar o resumo de tabelas, através do uso da linguagem natural. Seu nome é uma sigla para *Natural Language Processing for Spreadsheet Information Retrieval*.

O sistema permite que um usuário execute certas funções de extração de informações dizendo ao sistema o que eles gostariam à sua maneira. Ele permite ao usuário filtrar linhas ou colunas, para contar o número de linhas que atendem a um determinado conjunto de critérios. Também permite que os usuários gerem tabelas que contam o número de linhas que atendem a certos critérios [8].

A implementação do sistema usa uma interface baseada em texto onde um usuário digita no que eles gostariam de perguntar ao sistema. A resposta gerada pelo sistema é apresentada ao usuário por meio de uma caixa de alerta.

7. DEACON

Na proposta do DEACON, seu autor (Thompson) define uma linguagem formal e uma técnica para determinar o significado das frases nessa linguagem. A técnica semântica é usar regras de interpretação que definem ações (ou sequências de ações) envolvendo os objetos de um conjunto finito de categorias de estruturas de memória.

Para analisar uma frase, o sistema DEACON reconhece as cadeias de caracteres que a formam. Para fazer isso, um dicionário de termos de vocabulário é construído a partir de definições digitadas pelo usuário. Um termo do vocabulário pode ser uma palavra (uma cadeia de caracteres entre caracteres em branco em uma frase) ou um idioma (uma sequência de duas ou mais palavras, como São Francisco) [9].

Devido à natureza do programa, a maioria dos termos do vocabulário normalmente são objetos, suas características e suas inter-relações. No DEACON, esses termos denotam estruturas no banco de dados.

8. CONVERSE

O programa CONVERSE é uma aplicação que realiza análise sintática de frases em inglês de maneira complexa. O sistema é composto de duas partes:

1. Um compilador de linguagem natural que traduz sentenças declarativas, imperativas e interrogativas em inglês para declarações procedurais em uma linguagem intermediária formal.
2. Um sistema de gestão que aceita estas declarações procedurais e realiza operações de armazenamento, busca e inferência especificadas.

Primeiro, todas as palavras em uma frase de entrada são examinadas no dicionário. Então, o compilador emprega uma gramática de reconhecimento e uma rede de dados conceituais, caracterizando a natureza do discurso, para aplicar informações semânticas e sintáticas para a análise e interpretação da string de entrada [10].

9. LUNAR

O sistema Lunar Sciences Natural Language Information System (lsnlis) criado por Bolt Beranek e Newman durante 1972 e 1973, foi pioneiro no desenvolvimento de sistemas que respondem a perguntas para um banco de dados e tinha como finalidade criar um

interface que aceita comandos na língua inglesa para acessar os um banco de dados de informações sobre a lua, respondendo perguntas sobre a composição química da lua.

O projeto foi desenvolvido devido a dificuldade em acessar essas informações uma vez que as mesmas se encontram espalhadas em diversos artigos, para facilitar esse acesso, foi criado um sistema que juntaria essas informações. Após juntar todas as informações seria necessário criar um método de acesso a essas informações que poderia ser facilmente utilizado pelos cientistas e para isso se utilizaram de um sistema de linguagem natural que entenderia o que os pesquisadores estariam pedindo, saberia interpretar as informações e realizar cálculos e processos com esses dados. O formato semântico utilizado foi similar ao descrito por Woods(1967,1968), e a linguagem utilizada foi a LISP [11].

10. SHRDLU

O sistema SHRDLU foi desenvolvido por Terry Winograd(1968-1970) para estudar sobre linguagens naturais. O sistema foi escrito em MacLisp.

O programa realiza uma conversa entre o computador e o usuário sobre um mundo de blocos que pode ser interagido pela máquina.

Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.
Computer: OK.
Person: What does the box contain?
Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.
Person: What is the pyramid supported by?
Computer: THE BOX.
Person: How many blocks are not in the box?
Computer: FOUR OF THEM.
Person: Is at least one of them narrower than the one which i told you to pick up?
Computer: YES, THE RED CUBE.
Person: Is it supported?
Computer: YES, BY THE TABLE.
Person: Can the table pick up blocks?
Computer: NO.

Figura 4. Exemplo de perguntas feitas ao ASK

11. RANDEZVOUS

O RENDEZVOUS, um sistema experimental desenvolvido pela Codd na IBM Research em 1974. RENDEZVOUS opera transformando a consulta original em uma expressão de cálculo relacional, ao fazê-lo usa uma base de conhecimento. A transformação da consulta original é feita de modo fragmentário. De modo geral, poderá haver fragmentos que o RENDEZVOUS não entenda.

Para tanto ele utiliza um diálogo de esclarecimento com o usuário, tentando extrair os fragmentos que não foram entendidos. Além disso, RENDEZVOUS é mais apropriado quando a consulta é realizada distante fisicamente do sistema central, porque não há necessidade de acessar o banco de dados [12].

12. LADDER

O sistema LADDER foi projetado para fornecer informações sobre os navios da Marinha dos EUA. O sistema LADDER usa semântica gramática para analisar questões para consultar uma base de dados. Ele usa a técnica de gramáticas semânticas que intercala o processamento sintático e semântico. Analisando a entrada e mapeamento da árvore de análise para uma consulta de banco de dados a resposta da pergunta está concluída.

O primeiro componente do sistema é para Acesso Informal de Linguagem Natural à Marinha Dados (INLAND). Este componente aceita perguntas em um linguagem natural e gera uma consulta ao banco de dados. As consultas do INLAND são direcionadas ao Acesso Inteligente a Dados (IDA). Este componente que é o segundo componente do LADDER constrói um fragmento de um consulta ao IDA para cada unidade sintática de nível inferior no Língua Inglesa. Consulta de entrada e esses fragmentos são em seguida, combinados com unidades sintáticas de nível superior para serem reconhecidos.

Os fragmentos combinados são enviados como um comando para IDA no nível da frase. IDA iria redija uma resposta que seja relevante para o original do usuário consulta, além de planejar a sequência correta de arquivo consultas. O terceiro componente do sistema LADDER é para File Access Manager (FAM). O FAM encontra a localização dos arquivos genéricos e controla o acesso a eles no banco de dados distribuído. O LISP foi usado para implementar o sistema LADDER [13].

13. CHAT-80

Foi desenvolvido por Fernando Pereira em 1983 um sistema ILNBD, tinha como principal objetivo o tratar o problema de transportabilidade de interfaces entre domínios.

O sistema utilizava a linguagem Prolog para acessar o banco de dados do Prolog na área de geografia. O módulo do chat 80 possuía funções para acessar os dados disponíveis em arquivos e convertê-los em um formato aceito pelo modelo FOL[16].

14. TEAM (Transportable English database Access Medium)

TEAM (Transportable English Data Access Medium) é uma interface de linguagem natural transportável (NL) para um banco de dados. É uma ferramenta de considerável poder que permite ao usuário recuperar dados e obter respostas às perguntas, fazendo perguntas e dando comandos em inglês, em vez de uma linguagem de consulta formal. Além disso, o TEAM não se limita a nenhum banco de dados específico, mas pode ser adaptado para demonstrar a recuperação em linguagem natural em uma ampla variedade de domínios de aplicação.

Para alguém preocupado em escrever interfaces de linguagem natural (NLI), qualquer passo adiante no sentido de tornar tais interfaces transportáveis devem ser de grande

interesse, por isso, TEAM foi um marco em portabilidade. Foi planejada para ser facilmente configurada pelos administradores de banco de dados sem qualquer conhecimento sobre ILNBDs.

O ponto principal é que, ao adaptar o TEAM a um novo banco de dados, o nível de especialização não precisa incluir nenhum conhecimento especial sobre linguística, processamento n-1 ou NLI em particular. Durante o processo de aquisição orientado por menu, o chamado especialista em banco de dados (DBE) fornece apenas informações sobre a estrutura do banco de dados e seu domínio de assunto. O TEAM deve extrair qualquer informação linguística especial necessária das respostas que o DBE fornece sobre as frases de amostra. Transportabilidade, manter o sistema o mais sensível ao contexto possível tornou necessário o processamento sequencial de consultas (isto é, completar estruturas sintáticas antes da análise semântica) [14].

15. ASK

ASK(A simple knowledgeable system) é um sistema de ILNBD simples. O foco do ASK é criar um sistema para o usuário que deseja criar, testar, modificar, aumentar e utilizar um banco de dados próprio, um sistema para pesquisa, gestão ou escritório.

A simplicidade do sistema vem da sua implementação que diferente de outros sistemas de linguagem natural que interpretam frases completas o ASK utiliza apenas alguns pontos da frase para interpretar o significado como Classes, objetos, atributos e relacionamentos, isso permite a adição de novos termos ao banco de dados[16].

```
>How many ships are there?
 7
>What is known about ships?
 some are in the following classes:
     Navy
     freighter
     old
     tanker
 all have the following attributes:
     destination
     home port
 some have the following attributes:
     cargo
 all have the following number attributes:
     age
```

Figura 5. Exemplo de perguntas feitas ao ASK[15]

```
 some have the following number attributes:
     speed
     length
     beam
>List the destinations and home port of
 each ship.
 ship            destination     home port
 Ubu             New York        Naples
                 Tokyo           ---
 Maru            Oslo            Tokyo
 Kittyhawk       Naples          Boston
                 Boston          ---
                 London          ---
 Alamo           London          London
                 New York        ---
 North Star      London          New York
 Nimitz          London          Norfolk
 Saratoga        unknown         Norfolk
>What cities are the home ports of ships
 whose destination is London?
 Boston
 London
 New York
 Norfolk
>Are there ships that do not have a cargo?
 yes
>What is the number of New York ships?
 There are 2 answers:
 (1) New York (destination) ships
 2
 (2) New York (home port) ships
 1
>How many ships are there with lnegth
 greater than 600 feet?
 Spelling correction: "lnegth" to "length"
 4
>What ships that carry wheat go to London or
 Oslo?
           ships that carry wheat
 London    Maru
 Oslo      Alamo
>Does the Maru carry wheat and go to London?
 yes
```

Figura 6. Continuação de perguntas feitas ao ASK[14]

16. JANUS

O sistema JANUS foi desenvolvido pela BBN e ISI(Information Science Institute) em 1988 para a recuperação de navios da frota do Pacífico da força naval dos EUA. O sistema JANUS utiliza a arquitetura gramatical de Montague para elaborar sua semântica, a linguagem adotada é a inglesa e a arquitetura gramatical gera uma expressão lógica que é comparada com o banco de dados.

O primeiro nível converte a frase para expressão lógica chamada Expressões Semânticas Ambíguas (EFls) nesse nível o contexto é ignorado, mas um modelo de domínio é usado para evitar ambiguidades e um modelo de discurso converte advérbios de tempo, um passo importante para o sistema que depende de localizações e horários para calcular as rotas de navios, além de manter o conhecimento do que pronomes estão se referindo

Os dois modelos traduzem o EFL para uma Expressão Semântica não Ambígua(WML) que por sua vez é traduzida para uma expressão da língua de pergunta da base de dados [16].

17. TCL

O TCL é um interpretador para uma linguagem de comando de ferramenta. Ele consiste em uma biblioteca que contém ferramentas como editores e depuradores como interpretador de comandos.

Ele fornece (a) um analisador para um comando textual simples linguagem, (b) uma coleção de comandos de utilitários integrados e (c) uma interface C que as ferramentas usam para aumentar os comandos integrados com comandos específicos da ferramenta [17].

Referências

[1] Cantarelli, Elisa Maria Pivetta. Acesso a Base de Dados Através da Linguagem Natural. Frederico Westphalen, p.64, jul. 1998.

[2] Barbosa, Cinthyan Renata Sachs. Processamento de Linguagem Natural. Londrina, p.3, 01 set. 2021. Apresentação em slide. 3 slides. color. Acesso em: 06 set. 2021.

[3] Lindsay R. K. Inferential memory as the basis of machines which understand natural language. Edited by Edward A. Feigenbaum & Julian Feldman, Computers and Thought.

[4] Cantarelli, Elisa Maria Pivetta. Acesso a Base de Dados Através da Linguagem Natural. Frederico Westphalen, p.64-65, jul. 1998.

[5] Bobrow, Daniel G. "Natural language input for a computer problem solving system." (1964).

[6] Norvig, Peter (1992). Paradigms of artificial intelligence programming:case studies in Common Lisp. San Francisco, California: Morgan Kaufmann. pp. 109–149. ISBN 1-55860-191-0.

[7] Simmons, R.F. and McConlogue, K.L. (1963), Maximum-depth indexing for computer retrieval of English language data. Amer. Doc., 14: 68-73. https://doi.org/10.1002/asi.5090140111.

[8] Derek Flood, Kevin Mc Daid, & Fergal Mc Caffery (2009). NLP-SIR: A Natural Language Approach for Spreadsheet Information Retrieval.

[9] Craig, James A., et al. "Deacon: Direct english access and control." Proceedings of the November 7-10, 1966, fall joint computer conference. 1966.

[10] Kellogg, Charles, et al. "The CONVERSE natural language data management system: current status and plans." Proceedings of the 1971 international ACM SIGIR conference on Information storage and retrieval. 1971.

[11] Woods, William & Kaplan, Ronald & Webber, Bonnie. (1972). The Lunar Science Natural Language Information System: Final Report.

[12] Cantarelli, Elisa Maria Pivetta. Acesso a Base de Dados Através da Linguagem Natural. Frederico Westphalen, p.66-67, jul. 1998.

[13] Jadhav, Sneha. Natural Language to Database Interface. International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 2, p. 2, February 2014.

[14] Grosz, Barbara J. TEAM: an experiment in the design of transportablen natural-language interfaces. Volume 32, Edition 2, p.173-243, may. 1998.

[15] Bozena H. Thompson & Ferederic B. Thompson, Introducing ask, a simple knowledgeable systema.

[16] Agosti, Cristiano (2003), Interface em Linguagem Natural para Banco de Dados: uma abordagem prática.

[17] Ousterhout, John K. Tcl: An embeddable command language. University of California, Berkeley, Computer Science Division, 1989.