



Pós-Graduação em Ciência da Computação

Augusto Lazzarotto de Lima

Um Parser Semântico para Comunicação Aumentativa e Alternativa



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
<http://cin.ufpe.br/~posgraduacao>

Recife
2018

Augusto Lazzarotto de Lima

Um Parser Semântico para Comunicação Aumentativa e Alternativa

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: ciência da computação
Orientador: Robson do Nascimento Fidalgo
Coorientador: Rinaldo José de Lima

Recife
2018

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

L732p Lima, Augusto Lazzarotto de
 Um parser semântico para comunicação aumentativa e alternativa /
 Augusto Lazzarotto de Lima. – 2018.
 92 f.: il., fig., tab.

 Orientador: Robson do Nascimento Fidalgo.
 Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn,
 Ciência da Computação, Recife, 2018.
 Inclui referências.

 1. Banco de dados. 2. Ontologia. I. Fidalgo, Robson do Nascimento
 (orientador). II. Título.

 025.04 CDD (23. ed.) UFPE- MEI 2018-122

Augusto Lazzarotto de Lima

Um Parser Semântico para Comunicação Aumentativa e Alternativa

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 01/03/2018.

BANCA EXAMINADORA

Prof. Dr. Frederico Luiz Gonçalves Freitas
Centro de Informática/UFPE

Profa. Dra. Roberta Vilhena Vieira Lopes
Instituto de Computação/UFAL

Prof. Dr. Robson do Nascimento Fidalgo
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho a minha família, amigos e professores que me deram o suporte e auxílio necessário para chegar até aqui.

AGRADECIMENTOS

A presente dissertação de mestrado não teria sido concretizada sem o apoio de pessoas importantes na minha vida.

Em primeiro lugar, agradeço a minha namorada Fernanda, meu porto seguro, companheira de todas as horas, exemplo de ser humano e profissional. Pela sua compreensão, respeito, tolerância, por ter sacrificado tanto em prol de um objetivo maior e, por todas as atitudes que te faz merecedora do meu amor.

Muito obrigado, pai, Arlindo, que sempre acreditou no meu potencial, dando todo o apoio, suporte e carinho, incentivando para que atingisse meus objetivos e que mesmo de longe esteve presente neste processo. Deu certo!

Agradeço também aos meus pais e irmã herdados da minha namorada: Marcos, Silvia e Danielle. Muito obrigado por acreditarem em mim e fazerem aquela torcida de sempre! Obrigado Família Silva!

Não posso esquecer de agradecer ao meu amigo, professor e orientador, Robson, por ter confiado em um desconhecido, do sul do país, um trabalho tão grandioso. Além de auxiliar com a tua experiência, capacidade, orientação e principalmente a tua amizade, consegui finalizar este trabalho.

Não menos importante, agradeço meu co-orientador, Rinaldo, que com suas dicas fundamentais e suas pilhas de materiais para leitura ajudou para que este trabalho fosse concluído.

Aos meus queridos amigos e colegas de trabalho, Natalia, Edson e Thiago, muito obrigado. Vocês foram essenciais, me ajudando sempre que precisei: tecnicamente, emocionalmente e gastronomicamente hahaha

A todos que de alguma forma contribuíram para que este trabalho fosse realizado, meu mais sincero, muito obrigado!

RESUMO

Os Sistemas de Comunicação Aumentativa e Alternativa (SCAA), a partir da seleção de símbolos com legendas, ajudam a desenvolver a comunicação e a reduzir a frustração de quem tem dificuldade em falar (e.g. pessoas com Paralisia Cerebral, Síndrome de Down, Microcefalia e Autismo). Na educação especial, crianças em fase de alfabetização podem apresentar problemas de ordenamento das palavras, conhecimento reduzido de verbos, omissão de elementos gramaticais e dificuldade em estruturar frases compreensíveis. Neste contexto, entende-se que os SCAA devem oferecer pistas visuais para construir frases em ordem direta e bem formadas. Contudo, não foram encontrados trabalhos relacionados que oferecessem esta facilidade. A fim de superar esta limitação, neste trabalho apresenta-se a especificação e implementação de um parser semântico para SCAA que visa guiar, de forma visual e em tempo de execução, a criação de frases em ordem direta e bem formadas. O desenvolvimento desta proposta se inicia com o levantamento do estado da arte (i.e., fundamentação teórica e trabalhos relacionados), o que permite entender os principais conceitos relacionados, bem como as necessidades existentes no contexto dos SCAA. Na sequência, de modo a prover pistas visuais e a fornecer suporte para construir frases em ordem direta e bem formadas, a abordagem *Colorful Semantics* (CS) e a teoria de linguagem SLIM (*Surface compositional, time-Linear, Internal Matching*) foram adaptadas ao contexto dos SCAA. Por fim, com base nas pistas visuais do CS e na teoria de linguagem SLIM, faz-se uma avaliação sobre as diferentes possibilidades de compor uma frase em ordem direta e bem formada. Além disso, também se analisou a redução da quantidade de toques/cliques e do esforço cognitivo necessários para a seleção do símbolo desejado. O *parser* proposto é capaz de analisar 10 diferentes formas de frases em ordem direta e bem formadas, bem como reduz o número de toques/cliques e o esforço cognitivo necessários para construir uma frase compreensível. Esta proposta especificou e implementou um *Parser* Semântico para Comunicação Aumentativa e Alternativa (CAA) que guia, de forma visual e em tempo de execução, a construção de frases telegráficas bem formadas em ordem direta. Para isto, criou-se uma interface simples que alterna entre duas telas, que orientam o fluxo do funcionamento do *parser*, tornando a construção das frases mais fáceis e rápidas.

Palavras-chaves: Sistemas de Comunicação Aumentativa e Alternativa. *Parser* Semântico. *Colorful Semantics*. Teoria SLIM. Frases bem formadas.

ABSTRACT

The Augmentative and Alternative Communication Systems (AACs), by means of selection of symbols with captions, helps to develop the communication and reduce the frustration of those who have speech impairments (e.g., people with Cerebral Palsy, Down Syndrome, Microcephaly and Autism). In special education, children in literacy process can have problems with word ordering, reduced knowledge of verbs, omission of grammatical elements and difficulties in structuring comprehensive sentences. In this context, an AACs should provide visual clues for building well-formed sentences in a direct order. However, no related work was found that offers this facility. In order to overcome this limitation, this proposal presents the specification and implementation of a Semantic Parser for AACs that aims to visually guide, at runtime, the creation of well-formed sentences in a direct order. The development of this proposal begins with a survey of the state of art (i.e., theoretical foundation and related works), which allows the understanding of the main related concepts as well as the existing needs in the AACs context. Following, in order to provide visual clues and support to construct well-formed sentences in direct order, the Colorful Semantics (CS) approach and the SLIM (Surface Compositional, Time-Linear, Internal Matching) theory of language were adapted to AACs context. Finally, based on the visual cues of CS and SLIM theory, an evaluation of the different possibilities of composing a well-formed sentence in the direct order is made. In addition, it was analyzed the reduction of the amount of touches/clicks and cognitive effort required to select the desired symbol. The proposed parser is capable to analyze 10 different forms of well-formed sentences in the direct order. Furthermore, it has been shown that this reduces the number of touches/clicks and the necessary cognitive effort to construct an understandable sentence. This proposal specifies and implements a Semantic Parser for Augmentative and Alternative Communication (AAC) that visually guides, at runtime, the construction of well-formed telegraphic sentences in direct order. To accomplish this, a simple interface that alternates between two screens was created, which guide the flow of the semantic parser, making the construction of sentences easier and faster.

Key-words: Augmentative and Alternative Communication Systems. Semantic Parser. Colorful Semantics. SLIM Theory. Well-formed sentences.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de frase com pictogramas do ARASAAC (PALAO, 2017). . . .	15
Figura 2 – Exemplo de frase sintaticamente correta e semanticamente incorreta. .	15
Figura 3 – Prancha de Comunicação Alternativa de Baixa Tecnologia	20
Figura 4 – Prancha de Comunicação Alternativa de Alta Tecnologia	20
Figura 5 – Chave de Fitzgerald	21
Figura 6 – Padrão de cores do <i>Colourful Semantics</i> (CS)	21
Figura 7 – Exemplo de frase com o CS	22
Figura 8 – Componentes do aBoard	22
Figura 9 – Organização do Vocabulário aBoard	23
Figura 10 – Exemplo de construção de frase no aBoard	24
Figura 11 – Exemplo de <i>frame</i> e suas ligações semânticas	27
Figura 12 – Comparação entre as análises com base no princípio metodológico . .	30
Figura 13 – Ilustração da Teoria <i>Surface-compositionality time-Linearly & Internal Matching</i> (SLIM)	30
Figura 14 – Exemplo de regra gramatical <i>Left-Associative Grammar</i> (LAG)	31
Figura 15 – <i>Bottom-up Left-Associative</i>	32
Figura 16 – Exemplo da análise da frase “ <i>Mary gives Fido a bone</i> ” utilizando a LAG	32
Figura 17 – Exemplo de <i>proplet</i> e seus membros	34
Figura 18 – Exemplo da derivação LA-Hear	35
Figura 19 – Exemplo da navegação LA-Think	36
Figura 20 – Exemplo de derivação LA-Speak	36
Figura 21 – Diagrama Simplificado da Proposta	39
Figura 22 – Interface do Pictogrammar	40
Figura 23 – Fluxo de informação do sistema de criação do <i>Blissymbols</i>	41
Figura 24 – Arquitetura do Sistema de criação <i>Blissymbols</i>	42
Figura 25 – Interface da proposta de Netzer	43
Figura 26 – Hierarquia Anotada dos Substantivos	44
Figura 27 – Interface protótipo Sistema de Compressão-Expansão (COMPANSION)	45
Figura 28 – Interface do aBoard no cenário ideal	50
Figura 29 – Cenários de interação do usuário	51
Figura 30 – Visão geral do funcionamento do sistema para a construção da sentença “ <i>I drink cold water</i> ”	52
Figura 31 – Adaptação da Teoria SLIM	53
Figura 32 – Exemplo de arquivo de projeto <i>Java Surface Linear Internal Matching</i> (JSLIM)	54
Figura 33 – Exemplo de arquivo de atributos .att	55

Figura 34 – Exemplo de arquivo de léxico .all	56
Figura 35 – Diferença da estrutura de <i>proplet</i> original e proposto	56
Figura 36 – Arquivo de inserção do léxico e restrições sintáticas e semânticas no <i>Database Semantics</i> (DBS)	57
Figura 37 – Comparação entre o DBS original e o Adaptado	58
Figura 38 – Arquitetura Geral da Proposta	59
Figura 39 – Arquitetura do Gerador do Léxico Personalizado	60
Figura 40 – Arquitetura do <i>parser</i> semântico	62
Figura 41 – Exemplo de <i>frame</i> semântico na Ontologia	63
Figura 42 – Diagrama de Atividades do Gerador	64
Figura 43 – Exemplo de estrutura de verbos, substantivos e adjetivos	65
Figura 44 – Interface do simulador Sistema de Comunicação Aumentativa e/ou Al- ternativa (SCAA)	66
Figura 45 – Diagrama de Atividades Macro	67
Figura 46 – Diagrama de Atividades da Etapa A	68
Figura 47 – Exemplo de busca por candidatos da Opção “ <i>Who?</i> ”	69
Figura 48 – Diagrama de Atividades da Etapa B	70
Figura 49 – Exemplo de estrutura <i>template</i> e filtro de classe gramatical	71
Figura 50 – Exemplo de regras gramaticais do arquivo <i>syntax.rul</i>	71
Figura 51 – Exemplo da Etapa B	72
Figura 52 – Diagrama de Atividades da Etapa C	73
Figura 53 – Exemplo da Etapa C	74
Figura 54 – Exemplo das listas de <i>frames</i> do verbo restringidos pelo sujeito	74
Figura 55 – Exemplo do cenário atual	75
Figura 56 – Especialização da categoria “ <i>Nourishment</i> ” no aBoard	78
Figura 57 – Criação da frase “ <i>I drink water</i> ” no aBoard	79

LISTA DE TABELAS

Tabela 2	–	Critérios abrangidos pelo trabalho de MARTÍNEZ-SANTIAGO et al.	40
Tabela 3	–	Critérios abrangidos pelo trabalho de NETZER	43
Tabela 4	–	Critérios abrangidos pelo trabalho de PAHISA-SOLé	46
Tabela 5	–	Análise comparativa dos trabalhos relacionados	47
Tabela 6	–	Comparativo entre as frases geradas e a variação do CS	76
Tabela 7	–	Comparativo entre a quantidade de cliques do aBoard e da proposta	78
Tabela 8	–	Análise Comparativa entre os trabalhos relacionados e a proposta	81

LISTA DE ABREVIATURAS E SIGLAS

ASHA	<i>American Speech-Language-Hearing Association</i>
CAA	Comunicação Aumentativa e/ou Alternativa
CAT	Comitê de Ajudas Técnicas
COMLEX	COMLEX <i>Syntax Dictionary</i>
COMPANSION	Sistema de Compressão-Expansão
CS	<i>Colourful Semantics</i>
DBS	<i>Database Semantics</i>
DF	Descrição Funcional
EVCA	<i>English Verb Classes and Alternations</i>
FK	Chave de Fitzgerald
GF	<i>Grammatical Framework</i>
JSLIM	<i>Java Surface Linear Internal Matching</i>
LAG	<i>Left-Associative Grammar</i>
PCA	Prancha de Comunicação Alternativa
PLN	Processamento de Linguagem Natural
PP	Pergunta de Pesquisa
PS	<i>Parser Semântico</i>
SCAA	Sistema de Comunicação Aumentativa e/ou Alternativa
SLIM	<i>Surface-compositionality time-Linearity & Internal Matching</i>
SUpO	<i>Simple Upper Ontology</i>
SURGE	<i>Systemic Unification Realization Grammar of English</i>
TA	Tecnologia Assistiva

SUMÁRIO

1	INTRODUÇÃO	14
1.1	PROBLEMA DE PESQUISA E MOTIVAÇÃO	15
1.2	OBJETIVOS DA PESQUISA	16
1.3	DELIMITAÇÃO DO ESCOPO	17
1.4	CONTRIBUIÇÕES	17
1.5	ESTRUTURA DO DOCUMENTO	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	COMUNICAÇÃO AUMENTATIVA E/OU ALTERNATIVA	19
2.1.1	aBoard	22
2.2	ONTOLOGIA	24
2.3	PROCESSAMENTO DE LINGUAGEM NATURAL	27
2.4	TEORIA SLIM DA COMUNICAÇÃO EM LINGUAGEM NATURAL	28
2.4.1	<i>Left-Associative Grammar</i>	31
2.4.2	DB Semantics	33
3	TRABALHOS RELACIONADOS	37
3.1	CRITÉRIOS PARA A AVALIAÇÃO DOS TRABALHOS	37
3.2	APRESENTAÇÃO DOS TRABALHOS	37
3.3	ANÁLISE COMPARATIVA	46
4	PROPOSTA	49
4.1	VISÃO GERAL	49
4.2	ADAPTAÇÕES	53
4.2.1	Adaptação Científica	53
4.2.2	Adaptação Tecnológica	54
4.3	ARQUITETURA	58
4.3.1	Arquitetura Gerador do Léxico Personalizado	59
4.3.2	Arquitetura Parser Semântico	61
4.4	VISÃO FUNCIONAL	63
4.4.1	Visão Funcional do Gerador do Léxico Personalizado	63
4.4.2	Visão Funcional do <i>Parser</i> Semântico	65
4.5	AVALIAÇÃO PRÁTICA DO PROTÓTIPO	76
4.6	AVALIAÇÃO COMPARATIVA DO PROTÓTIPO	80
4.7	CONSIDERAÇÕES FINAIS DO CAPÍTULO	81
5	CONCLUSÃO	83

5.1	CONSIDERAÇÕES FINAIS	83
5.2	CONTRIBUIÇÕES	85
5.3	LIMITAÇÕES	87
5.4	TRABALHOS FUTUROS	87
	REFERÊNCIAS	89

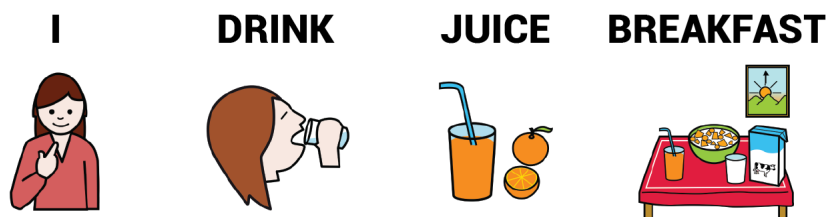
1 INTRODUÇÃO

Existem vários fatores que podem limitar, de forma temporária ou permanente, a capacidade de comunicação de uma pessoa, entre eles, a deficiência intelectual (e.g., Transtorno do Espectro Autista e Síndrome de Down) e as desordens neurológicas (e.g., Paralisia Cerebral, Lesão Crânio Encefálica e Microcefalia). A comunicação, em suas diversas formas, é uma necessidade básica que o ser humano tem de se socializar, expressar ideias, desejos e sentimentos e sua importância é destacada por diversos autores (TOMASELLO, 2009; LÁNYI C S E BACSA, 2004). Quando a privação na comunicação ocorre nos primeiros anos de vida, esta pode provocar déficit no desenvolvimento cognitivo e atraso na aquisição da língua escrita (MOUSINHO RENATA E SCHMID, 2008). No contexto da educação, os desafios surgem quando um educador tenta se comunicar com uma criança que tem limitações significativas na comunicação (SPEARS; TURNER, 2010), uma vez que os problemas na compreensão da linguagem também incluem a incapacidade de entender direções, perguntas ou comandos simples (LIM, 2011). De acordo com (BOLDERSON et al., 2011), estas crianças falham em produzir frases consistentes e inteligíveis pois possuem conhecimento reduzido de verbos e, geralmente, omitem elementos gramaticais da frase. Para estes casos, faz-se necessária a inserção de mecanismos de Tecnologia Assistiva (TA) que visam complementar a capacidade de comunicação destes indivíduos. Estes mecanismos são denominados dispositivos de Comunicação Aumentativa e/ou Alternativa (CAA).

A CAA é destinada a atender pessoas sem fala ou escrita funcional ou que estão em defasagem entre sua necessidade comunicativa e sua habilidade em falar e/ou escrever. Os recursos de CAA visam reconhecer e valorizar todas as tentativas de comunicação de um indivíduo – sejam elas por meio de gestos, expressões faciais, olhares, escrita ou desenhos –, e tentam complementá-las. Neste contexto, o SCAA é um recurso adequado para atender às necessidades básicas de comunicação dessas pessoas, pois usa símbolos pictográficos para representar conceitos, ações, objetos, etc. Assim, a comunicação é estabelecida por meio da seleção de uma sequência destes símbolos pictográficos. Na Figura 1 apresenta-se um exemplo de frase telegráfica construída com pictogramas. Neste contexto, uma frase telegráfica pode ser entendida como uma fala reduzida onde tudo o que não é essencial, em termos de palavras e flexões, é omitido (e.g., *"I drink juice breakfast"* é uma frase telegráfica para a frase *"I drink orange juice on the breakfast"*).

O principal uso dos SCAA é para a comunicação básica (i.e., socializar, expressar ideias, desejos e sentimentos) e, neste caso, a sintaxe e a semântica das frases produzidas não são uma prioridade. Assim, uma frase do tipo *"I cake eat"*, apesar de apresentar erros sintáticos, consegue comunicar o desejo de comer bolo, bastando para isso, a interpretação do interlocutor. Por outro lado, a frase *"I drink cake"* (cf. Figura 2), apesar de estar sintaticamente correta, apresenta erro semântico e, dada a sua ambiguidade, necessita

Figura 1 – Exemplo de frase com pictogramas do ARASAAC (PALAO, 2017).



Fonte: do Autor.

de uma interpretação um pouco mais cuidadosa por parte do interlocutor, uma vez que não se sabe se a intenção é beber alguma coisa que não seja o bolo, beber alguma coisa com bolo ou comer bolo. Além da comunicação, os SCAA podem ser vistos como uma ferramenta de apoio para a educação e para o desenvolvimento cognitivo de seus usuários. Assim, a qualidade das frases produzidas também deve ser levada em consideração. Neste contexto, a criação de frases sintática e semanticamente corretas pode ser estimulada a partir da combinação de um guia visual que orienta na sintaxe (i.e., as regras de boa formação de uma língua) com a sugestão de pictogramas considerando a sintaxe e a semântica (i.e., o sentido das palavras) entre os constituintes de uma frase. Assim, os SCAA associados a uma ferramenta de sugestão e validação que considere a sintaxe e semântica dos pictogramas pode servir de apoio para o desenvolvimento das habilidades linguísticas e da alfabetização do usuário.

Figura 2 – Exemplo de frase sintaticamente correta e semanticamente incorreta.



Fonte: do Autor.

1.1 PROBLEMA DE PESQUISA E MOTIVAÇÃO

O problema de pesquisa abordado nesta proposta é a sugestão de pictogramas para a construção de frases telegráficas bem formadas e com sentido no contexto da comunicação funcional. Nesta proposta considera-se *frase telegráfica bem formada e com sentido* aquela frase que representa uma fala reduzida, a qual está sintática e semanticamente correta. Na Figura 2 apresenta-se um exemplo de frase bem formada (i.e., sintaticamente correta de acordo com alguma regra gramatical), porém, sem sentido (i.e., semanticamente incorreta), uma vez que "cake" não é um complemento esperado para o verbo "to drink".

Destaca-se que a criação de frases erradas e sem sentido não favorecem ao desenvolvimento das habilidades linguísticas e à alfabetização dos usuários de SCAA.

Dado o exposto, este trabalho é motivado pelos seguintes fatos:

- SCAA permitem a criação livre de frases e não restringem o uso de pictogramas, o que pode gerar frases com erros sintáticos e semânticos;
- A criação de frases com erros sintáticos e semânticos não favorece ao desenvolvimento das habilidades linguísticas e à alfabetização do usuário;
- A sugestão sintática e semântica de pictogramas pode guiar o usuário na construção de frases corretas;
- A sugestão sintática e semântica de pictogramas pode aumentar o poder cognitivo (i.e., uso de recursos psicológicos como memória, atenção e raciocínio na resolução de problemas) do usuário na tentativa de se comunicar;
- A sugestão sintática e semântica de pictogramas pode aumentar a taxa de comunicação (i.e., velocidade) do usuário na tentativa de se comunicar;
- A sugestão sintática e semântica de pictogramas pode facilitar a comunicação do usuário de SCAA.

1.2 OBJETIVOS DA PESQUISA

O objetivo deste trabalho é desenvolver um *parser* semântico que facilite a comunicação e guie os usuários de SCAA, por meio de uma gramática simplificada e da sugestão de pictogramas, na criação de frases telegráficas bem formadas e com sentido. Para alcançar esse objetivo geral, tem-se como objetivos específicos:

- Propor e avaliar um método de interação para SCAA que guie o usuário na criação de frases bem formadas e com sentido;
- Propor e avaliar a sugestão de constituintes de uma frase criada pelo método de interação;
- Propor e avaliar um *Parser* Semântico que valide a sintaxe e a semântica das frases criadas a partir do método de interação para SCAA.

Dado o exposto, este trabalho visa responder a seguinte pergunta geral de pesquisa:

Pergunta de Pesquisa (PP): *O uso de um Parser Semântico aliado a um método de interação para SCAA pode guiar a criação de frases telegráficas bem formadas e com sentido, facilitando a comunicação do usuário em termos de aumento da taxa de comunicação e redução do esforço físico e cognitivo?*

De forma específica, os resultados serão agrupados visando responder as seguintes perguntas específicas:

- **(PP1)** *Como organizar a interface gráfica de um SCAA e a sua forma de interação de forma a guiar o usuário na criação de frases bem formadas e com sentido?*
- **(PP2)** *Como especificar um Parser Semântico para SCAA que facilite a criação de frases telegráficas bem formadas e com sentido?*

1.3 DELIMITAÇÃO DO ESCOPO

Esta proposta está delimitada em 3 aspectos:

1. **Língua** – dada a disponibilidade dos recursos léxicos e semânticos, este *parser* semântico para CAA é desenvolvido na língua inglesa devido a gama de recursos existentes para este idioma;
2. **Nível Linguístico** – esta proposta considera apenas a construção de frases telegráficas (cf. Seção 1.1);
3. **Sistema de CAA** – para fins de exemplificação, esta proposta utiliza a plataforma aBoard (LIMA et al., 2017), a qual é formada pelos *softwares* aBoard (um aplicativo para dispositivo móvel) e pelo *aBoard Editor* (um sistema *Web*). Na Seção 2.1.1 apresenta-se uma visão geral desta plataforma.

1.4 CONTRIBUIÇÕES

Este trabalho provê uma forma de guiar os usuários de SCAA, por meio de uma gramática visual simplificada e da sugestão de pictogramas, na construção de frases telegráficas bem formadas e com sentido. Assim, espera-se diminuir o esforço físico e influenciar no poder cognitivo do usuário, bem como aumentar o seu poder de expressão, reduzindo a quantidade de cliques e o tempo necessário para a construção de frases. Ressalta-se que, apesar de aplicado ao contexto de CAA, o conhecimento adquirido no decorrer deste trabalho pode ser aplicado em *Parsers* semânticos de outros domínios. Assim, de forma a deixar um legado para futuras aplicações, com este trabalho esperam-se as seguintes contribuições:

- Método de interação com a interface que guia o usuário na construção de frases bem formadas e com sentido;
- *Parser* semântico para a validação sintática e semântica de frases telegráficas no contexto de CAA.

1.5 ESTRUTURA DO DOCUMENTO

O restante deste documento é dividido em 4 capítulos. No Capítulo 2 apresenta-se os fundamentos teóricos necessários para embasar este trabalho, os quais são divididos em 4 seções principais: Comunicação Aumentativa e Alternativa, *Colourful Semantics*, Processamento de Linguagem Natural, Ontologia e Teoria SLIM da comunicação em linguagem natural. No Capítulo 3 apresenta-se os trabalhos relacionados a esta proposta. No Capítulo 4 apresenta-se o *Parser* Semântico para CAA, com uma visão arquitetural e outra aplicada, bem como a avaliação da proposta. No Capítulo 5 apresentam-se as considerações finais, as contribuições e as limitações deste trabalho, bem como direcionamentos para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo abordam-se os conceitos de Tecnologia Assistiva e Comunicação Aumentativa e Alternativa, *Colourful Semantics* (BRYAN, 2003), Processamento de Linguagem Natural, Ontologia, a teoria SLIM (HAUSSER, 1989). Estes conceitos formam a fundamentação teórica que servem de base para o entendimento da pesquisa.

2.1 COMUNICAÇÃO AUMENTATIVA E/OU ALTERNATIVA

Segundo o Comitê de Ajudas Técnicas (CAT) (Brasil, 2009), Tecnologia Assistiva (TA) é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social.

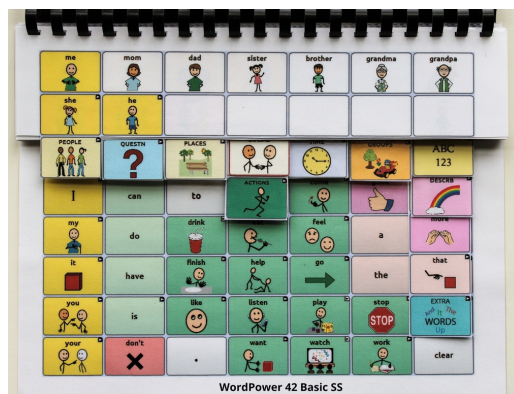
Neste contexto, TA refere-se a qualquer recurso ou procedimento que ofereça independência, autonomia e inclusão social e qualidade de vida ao usuário final. Conforme alguns exemplos citados em (BERSCH, 2008), a TA abrange projetos arquitetônicos para acessibilidade (e.g., rampas, adaptações em banheiros, elevadores), órteses e próteses (e.g., talas, apoios), recursos que permitem a comunicação expressiva de pessoas sem fala ou com limitações na mesma (e.g., prancha de comunicação alternativa de baixa e alta tecnologia). Neste sentido, tudo que facilite a vida de uma pessoa com deficiência de alguma forma é uma TA.

No que se refere a comunicação, a CAA é uma área da TA que se destina especificamente à ampliação de habilidades de comunicação. Para (BERSCH, 2008), a comunicação alternativa destina-se a pessoas sem fala ou sem escrita funcional ou em defasagem entre sua necessidade comunicativa e sua habilidade de falar e/ou escrever. Na definição da *American Speech-Language-Hearing Association* (ASHA) (ASHA, 2017), a CAA é a área da prática clínica que tenta compensar dificuldades ou incapacidades demonstradas, temporária ou permanentemente, por indivíduos com distúrbios graves de expressão comunicativa. Diante disto, as abordagens para CAA visam reconhecer, valorizar e complementar todas as formas de comunicação de um indivíduo (e.g., expressões faciais, gestos, olhares, escrita ou desenhos). Dentre as soluções, destacam-se o Sistema Braille – utilizado por deficientes visuais; a Língua Brasileira de Sinais (LIBRAS) – utilizada pelos surdos; e a Prancha de Comunicação Alternativa (PCA) – utilizada por pessoas sem fala funcional.

A PCA utiliza como vocabulário um conjunto de pictogramas que podem ser organizados por categorias (e.g., a categoria “bebidas” engloba os pictogramas “água” e “refrigerante”) ou por cores e exibidos em uma área de conteúdo. Vale destacar que existem

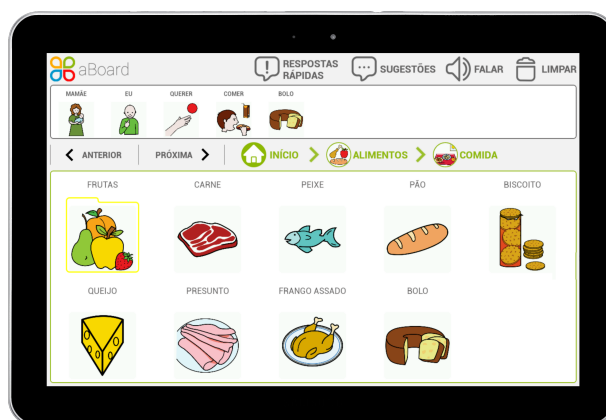
PCA de baixa tecnologia (cf. Figura 3) – pranchas baseadas em cartões de papel – e alta tecnologia (cf. Figura 4) – pranchas baseadas em *tablets* e *smartphones*. A comunicação na Figura 4 é estabelecida por meio da seleção de uma sequência de pictogramas, os quais são visíveis pelo usuário em uma área de criação de frase.

Figura 3 – Prancha de Comunicação Alternativa de Baixa Tecnologia



Fonte: (Saltillo, 2017)

Figura 4 – Prancha de Comunicação Alternativa de Alta Tecnologia

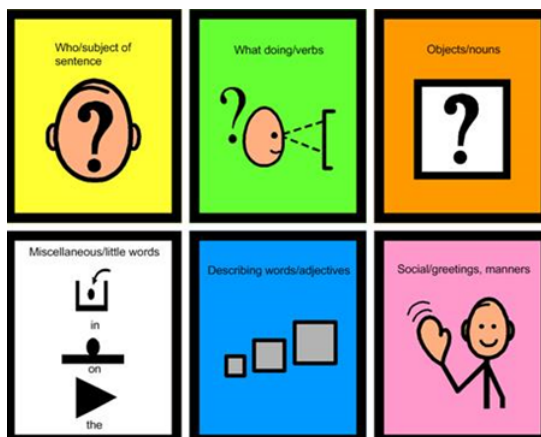


Fonte: (Assistive, 2017)

Com relação à organização do vocabulário, este é importante na especificação de dispositivos de CAA, uma vez que deve-se considerar os diversos contextos em que os usuários podem estar presentes. Para isto, os mediadores (i.e., familiares ou profissionais da saúde/educação) responsáveis pela organização podem agrupar este vocabulário por domínio ou por cores. No que diz respeito à organização por domínio, este está relacionado ao conceito do pictograma (i.e., o que ele representa) ou ao seu contexto de uso (i.e., onde ele é usado). Para as cores, faz-se uso de um padrão de cores que agrupam os pictogramas por cores que podem remeter às classes gramaticais e papéis semânticos. Dentre os padrões de cores mais conhecidos estão a Chave de Fitzgerald (FK) (FITZGERALD, 1949) e a CS

(BRYAN, 2003). O padrão de Fitzgerald foi originalmente criado para auxiliar crianças surdas a ler. Este padrão faz uso de um guia visual que originalmente referencia cores as principais classes gramaticais das palavras. Conforme a Figura 5, a paleta de cores é definida por: amarelo para pronomes, verde para verbos, laranja para os substantivos, azul para adjetivos, rosa para expressões sociais e branca para miscelânea (i.e., demais expressões).

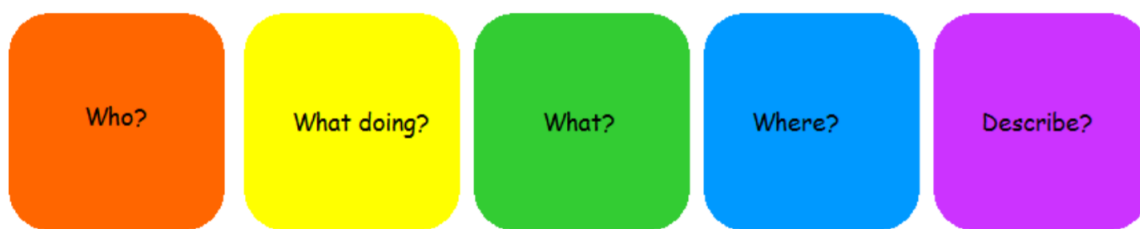
Figura 5 – Chave de Fitzgerald



Fonte: (Mayer-Johnson, 2017).

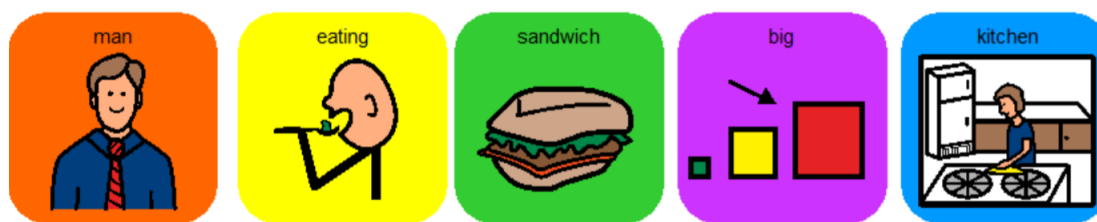
O padrão CS tem origem na necessidade de auxiliar crianças com dificuldades no planejamento de sentenças, na ordenação e recuperação de palavras (BOLDERSON et al., 2011). Como pode ser observado na Figura 6, este padrão dispõe de cinco níveis com diferentes cores para representar as *Wh-Questions*: *Who* (i.e., agente), *What Doing* (i.e., verbo), *What* (i.e., tema), *Where* (i.e., lugar) e *Describe* (i.e., descrição). Destaca-se que este padrão pode adaptar a posição do conceito “*Describe?*” para antes de “*Who?*”, depois de “*What Doing?*” e antes de “*Where?*”. Diante desta forte representação sintática e semântica, o *cs* funciona como um guia visual para organizar a frase, assim como para identificar os papéis semânticos de cada constituinte desta frase. Na Figura 7 apresenta-se o exemplo da frase “*The man is eating a sandwich in the big kitchen*”.

Figura 6 – Padrão de cores do CS



Fonte: (Integrated Treatment Services, 2017).

Figura 7 – Exemplo de frase com o CS

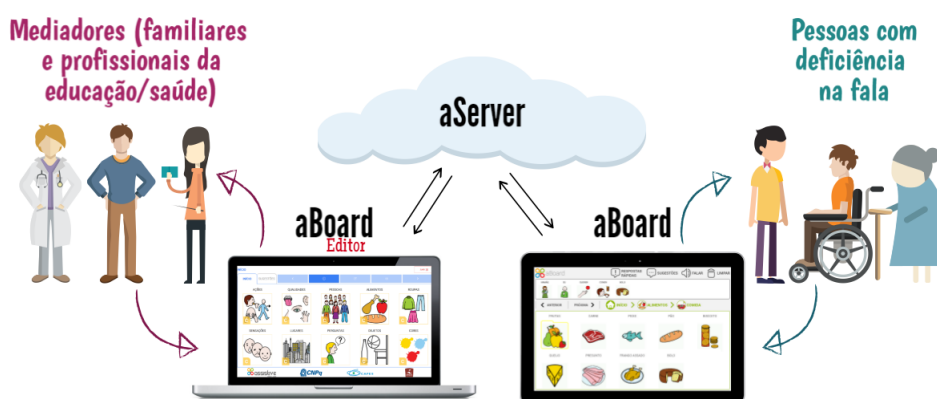


Fonte: (Integrated Treatment Services, 2017).

2.1.1 aBoard

Segundo (LIMA et al., 2017; FRANCO et al., 2017a), o aBoard é um SCAA de alta tecnologia em nuvem, que se adéqua as necessidades das pessoas com deficiência na fala. Este SCAA é composto por três componentes (cf. Figura 8): *aBoard*, *aBoard Editor* e *aBoard Server*. O *aBoard* é um aplicativo para dispositivo móvel que permite que pessoas com deficiência na fala (e.g., pessoas com Transtorno do Espectro Autista, Paralisia Cerebral, Microcefalia), consigam se comunicar a partir da seleção de pictogramas, produzindo frases que expressam seus desejos, opiniões e sentimentos. O componente *aBoard Editor* é um sistema *Web* que permite ao usuário mediador (e.g., familiares e profissionais de educação/saúde), responsáveis pelas pessoas com deficiência na fala, personalizar o conteúdo existente no aplicativo *aBoard*. Esta personalização refere-se a criação, edição e exclusão dos pictogramas que formam o conteúdo. O último componente, *aBoard Server* diz respeito ao serviço em nuvem que permite armazenar, disponibilizar e controlar o acesso aos dados personalizados.

Figura 8 – Componentes do aBoard

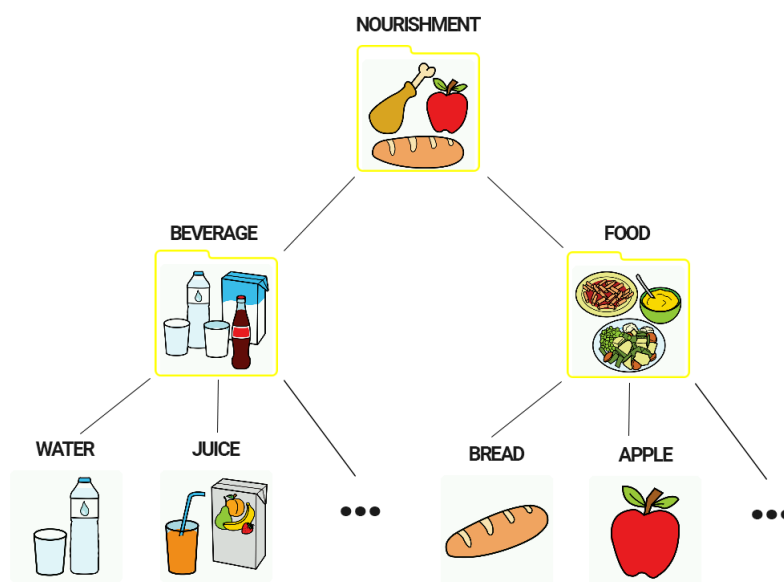


Fonte: (LIMA et al., 2017; FRANCO et al., 2017a).

Esta solução organiza o vocabulário a partir dos conceitos de categorias e elementos. Estes remetem a ideia de pastas e arquivos de um computador, ou seja, assim como os arquivos são organizados em pastas, os elementos são organizados em categorias. Desta

forma, a estrutura do vocabulário pode ser vista como uma hierarquia que pode ser especializada a cada nível. Por exemplo, dado a categoria “alimentos”, esta pode ser populada por elementos ingestíveis ou por outras categorias mais especializadas, como “comidas” e “bebidas”. Neste sentido, as categorias especializadas se tornam subcategorias de “alimentos” e que podem ser populadas por elementos que caracterizam alimentos líquidos para “bebidas” e sólidos para “comidas”. Além desta organização, existem duas categorias especiais denominadas “ações” e “qualidades”, estas organizam os elementos verbos e adjetivos, respectivamente. Na Figura 9 apresenta-se um exemplo de organização do vocabulário para a categoria “alimentos”.

Figura 9 – Organização do Vocabulário aBoard



Fonte: do Autor.

A comunicação é realizada por meio da seleção dos pictogramas existentes no vocabulário personalizado, assim, forma-se uma sequência de pictogramas que são vocalizados pelo aplicativo. Para melhor compreender o funcionamento da comunicação, apresenta-se na Figura 10 o exemplo da criação da frase “*Eu comer bolo*”. A primeira ação a ser feita é selecionar a categoria “pessoas”, em seguida, busca-se pelo pictograma “*Eu*” e o seleciona. Após, retorna-se à tela inicial do aplicativo para a escolha da palavra seguinte. Para isso é necessário escolher a categoria especial “ações”, onde será possível encontrar o verbo “*comer*”. Selecionando este pictograma, basta encontrar o próximo desejado. Deste modo, o usuário necessita voltar para a tela inicial e clicar na categoria “alimentos”, em seguida clicar na categoria “comidas” para enfim encontrar o pictograma “*bolo*”. Ao final, com a frase desejada completa, basta clicar no ícone para “falar” e a frase será vocalizada por um sintetizador de voz presente no *aBoard*.

Figura 10 – Exemplo de construção de frase no aBoard



Fonte: do Autor.

2.2 ONTOLOGIA

O termo Ontologia vem de um ramo da filosofia que trata da natureza, realidade e a existência dos entes (BORST, 1997). Na Ciência da Computação este termo foi introduzido por pesquisadores da área de inteligência artificial que trabalhavam com modelos computacionais com algum tipo de raciocínio automatizado. A partir dos anos 90, as ontologias começaram a ser tratadas como parte de sistemas baseados em conhecimento. Em sua primeira definição, (GRUBER, 1993) afirmou que a ontologia é uma especificação explí-

cita de uma conceitualização. Alguns anos depois, (BORST, 1997) definiu ontologia como uma especificação formal de uma conceitualização compartilhada e, finalmente, (STUDER; BENJAMINS; FENSEL, 1998) misturou as duas definições em “uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada”.

Na interpretação de (STUDER; BENJAMINS; FENSEL, 1998), “conceitualização” refere-se a um modelo abstrato de algum fenômeno no mundo ao ter identificado os conceitos relevantes desse fenômeno; “explícito” significa o tipo de conceito utilizado e as restrições de sua utilização são explicitamente definidas; “formal” refere-se ao fato de que a ontologia deve ser interpretável pela máquina, o que exclui a linguagem natural; e “compartilhado” reflete a noção de que uma ontologia capta conhecimento consensual, isto é, aceito por um grupo e não privado a um indivíduo.

Apesar de existirem diferentes definições para a ontologia, o principal propósito é permitir o compartilhamento, a reutilização de conhecimento e evitar ambiguidade de entendimento. De maneira geral, as ontologias são um conjunto de conceitos e suas inter-relações as quais fornecem uma perspectiva particular sobre determinado mundo, desenvolvidas para auxiliar na tomada de decisões de mecanismos que envolvem um processamento considerável de dados de forma inteligente.

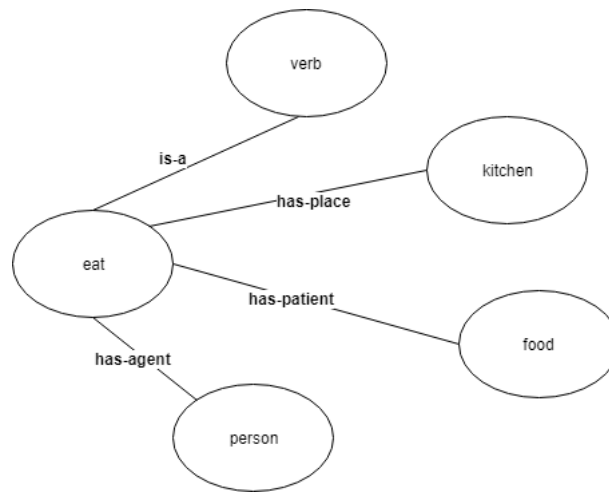
Para (GUARINO, 1998), existem diferentes tipos de ontologias conforme seu nível de generalidade:

- **Ontologia Genérica ou de Topo:** Descrevem conceitos gerais abstraindo os aspectos do mundo, como tempo, espaço, evento, etc., independentes de um problema específico ou domínio particular;
- **Ontologia de Domínio:** Descrevem, respectivamente, o vocabulário relacionado ao domínio genérico (e.g., medicina, biologia) ou tarefas/atividades genéricas (e.g., diagnóstico ou venda), especializando os termos introduzidos pela ontologia de topo;
- **Ontologia de Tarefa:** Descrevem tarefas ou atividades genéricas, expressando conceituações sobre a resolução de problemas, independentemente do domínio em que ocorram. Por exemplo, descrevem o vocabulário relacionado a uma atividade ou tarefa genérica, como diagnóstico ou processos de venda;
- **Ontologia de Aplicação:** Descrevem conceitos que dependem de um domínio particular e de uma tarefa específica. Estes conceitos são especializações dos termos das ontologias de domínio e de tarefa, que, normalmente correspondem a regras aplicadas a entidades de domínio enquanto executam determinada tarefa;
- **Ontologia de Representação:** Descrevem as conceituações que fundamentam os formalismos de representação de conhecimento, procurando tornar claros os compromissos ontológicos embutidos nestes formalismos.

No contexto desta proposta, serão detalhadas as ontologias de domínio e de representação linguística. Para a ontologia de domínio, representa-se o mundo real no que se refere ao conceito e as relações das entidades envolvidas em um domínio específico (e.g., pizzaria, doenças). Dado o exemplo da ontologia da pizza (HORRIDGE et al., 2006), uma pizza é formada por uma base (i.e., massa) e um topo (i.e., cobertura). A base é definida por uma crosta fina, grossa, macia e etc., e um topo pode conter uma cobertura com tomate, muçarela, carne e etc. Assim, a relação (e.g., *has_topping*, *has_base*) entre as bases e os tipos de topos formam diferentes tipos de pizzas (e.g., margherita, calabresa, vegetariana). Neste sentido, um sistema que utilize o conhecimento desta ontologia consegue identificar a composição (i.e., massa e cobertura) de uma pizza margherita, assim como identifica quais pizzas possuem massa fina ou quais utilizam o ingrediente tomate em sua cobertura.

Para a ontologia de representação linguística, segundo (DAHLGREN, 1995), esta retrata a “visão do mundo” codificado em linguagem natural, em diferentes níveis: morfológico, sintático e semântico. De modo geral, uma ontologia linguística combinada com uma ontologia de domínio pode conter conhecimento de significado, forma da palavra, flexões, *hiperonímia* (superclasse), *hiponímia* (subclasse), *meronímia* (parte-todo), *holonímia* (todo-parte), *antonímia* (sentidos opostos) e *similaridade* (sentidos similares). Por exemplo, dada a palavra “eat”, esta pode ser organizada na ontologia pertencente à uma classe que remete um verbo (i.e., classe gramatical), este verbo pode possuir um conjunto de relacionamentos que definem com que outras classes gramaticais este pode se relacionar. Este conhecimento pode ser definido utilizando a ideia de *frames* (BAKER; FILLMORE; LOWE, 1998), para isto, os termos são organizados em situações (i.e., *frames*), seus participantes (i.e., *frame elements*) e propriedades envolvidas. Os sentidos das palavras são estabelecidos por meio de uma unidade léxica (i.e., lemmas¹), que pode ser vista como uma ligação entre um termo e um *frame* (cf. Figura 11). O conhecimento linguístico pode ser utilizado por diversos tipos de ferramentas da área Linguagem Natural (e.g., *Parser Semântico*, *Geração de Linguagem Natural*, *Análise de Sentimentos*), uma vez que oferecem conhecimento linguístico necessário para resolver problemas específicos da língua.

¹ A forma canônica de uma palavra.

Figura 11 – Exemplo de *frame* e suas ligações semânticas

Fonte: do Autor.

2.3 PROCESSAMENTO DE LINGUAGEM NATURAL

A compreensão de uma frase em linguagem natural para o ser humano é tão fácil que quase nunca se pensa nisso. No entanto, para as máquinas, a tarefa de interpretar a linguagem natural especialmente entendimento do significado além do conteúdo literal, provou ser extremamente difícil e requer uma grande base de conhecimento. Segundo (CHOWDHURY, 2003), Processamento de Linguagem Natural (PLN) é um campo da Ciência da Computação que se concentra na compreensão e modelagem de linguagem humana, isto é, uma área de pesquisa que permite aos computadores entender e manipular textos de linguagem natural ou fala.

Conforme (JURAFSKY; MARTIN, 2009) um sistema de PLN é abordado da seguinte forma:

- **Fonética e Fonologia:** Estuda a linguística dos sons;
- **Morfológica:** Estudo das palavras de uma frase de forma independente, a fim de se obter um entendimento sobre sua forma primitiva;
- **Sintática:** O estudo das relações estruturais entre as palavras, ou seja, as regras que definem a construção das frases;
- **Semântica:** O estudo do significado, isto é, o processo de mapeamento das frases de uma linguagem com o objetivo de representar seu significado baseado nas construções sintáticas;
- **Pragmática:** O estudo de como a linguagem é usada para atingir objetivos, em outras palavras, como a linguagem é usada para se comunicar diante dos objetos e seu contexto;

- **Discursiva:** O estudo de unidades linguísticas maiores do que um único enunciado.

Os níveis de PLN acima refletem um tamanho crescente na unidade de análise, além de aumentar a complexidade e a dificuldade à medida que passamos do nível fonológico para o discursivo. Segundo (AKERKAR; JOSHI, 2008), quanto maior for a unidade de análise (i.e., do morfema à palavra, à frase, ao parágrafo e ao documento completo), maiores são os desafios de análise. Além disso, os níveis mais altos dependem da análise dos níveis mais baixos para a compreensão linguística.

No que tange ao nível sintático, a análise sintática (em inglês, *syntactic parser*) é a tarefa de reconhecer uma oração e atribuir-lhe uma estrutura sintática, baseada em uma gramática (JURAFSKY; MARTIN, 2009). Neste sentido, durante o *parsing*, busca-se por constituintes² que compõem ou irão compor uma frase ou oração. Nesta análise existem abordagens de busca para chegar às folhas de uma árvore de decisão (JURAFSKY; MARTIN, 2009):

- **Top-Down** – Um analisador *top-down* procura por uma árvore de análise tentando construir a partir do nó raiz até as folhas;
- **Bottom-Up** – Um analisador *bottom-up* começa com as palavras de entrada e tenta construir árvores a partir das palavras (folhas) para cima, aplicando regras de gramaticais uma de cada vez.

O nível semântico aborda o significado na linguística computacional, representando o significado de modo formal, capturando o conteúdo relacionado ao significado das entradas linguísticas. Estas representações destinam-se a preencher a lacuna da linguagem com o conhecimento do mundo comum (JURAFSKY; MARTIN, 2009). Em outras palavras, a análise semântica (em inglês, *semantic parser*), tem como objetivo identificar o significado literal das palavras de entrada, tentando encontrar uma relação entre as palavras de modo que a frase tenha um significado. Diante disto, um *parser* semântico se torna essencial para construir e analisar a estrutura semântica das palavras. Sua aplicabilidade pode ser encontrada em Sistemas de CAA, Sistemas COMPANSION³, sistemas de pergunta e resposta, geração de linguagem natural e etc.

2.4 TEORIA SLIM DA COMUNICAÇÃO EM LINGUAGEM NATURAL

Esta teoria destina-se a descrever o uso da linguagem humana como base para sistemas computacionais de linguagem natural. Esta teoria é construída com base em quatro

² Um constituinte é uma palavra, uma categoria lexical ou frasal que foi encontrada em uma sentença que está sendo analisada, ou uma que está sendo procurada e ainda não foi encontrada.

³ Compresão-Expansão (PENNINGTON; MCCOY, 1998)

princípios que podem definir os sistemas computacionais de linguagem natural: *Surface-compositionality* (i.e., princípio metodológico), *time-Linear* (i.e., princípio empírico), *Internal* (i.e., princípio ontológico) e *Matching* (i.e., princípio funcional). Estes princípios formam o que (HAUSSER et al., 2004) chama de uma certa concepção intuitiva da comunicação linguística natural, denominada SLIM. Esta teoria foi fundamentada com base em sistemas orientados a agentes, o que durante toda sua explicação deixa claro sua aplicabilidade nesta área.

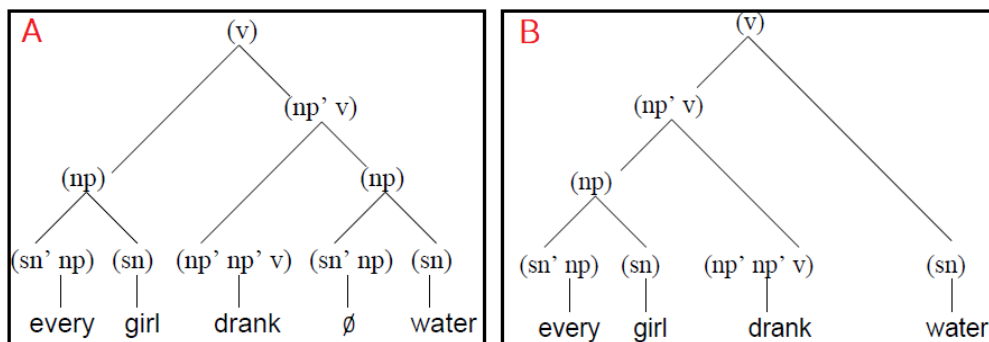
Para o princípio metodológico da *Surface-compositionality*, (HAUSSER, 2006) afirma que uma análise gramatical é composta de superfície se ela usa apenas as formas de palavras concretas como blocos de construção das composições. Neste sentido, todas as propriedades sintáticas e semânticas de uma expressão complexa derivam sistematicamente da categoria sintática e do significado literal dos itens léxicos. Assim, sua abordagem define que as formas superficiais das palavras são traduzidas (sintetizadas) em representações lógicas correspondentes de significado. Para melhor compreender, (HAUSSER et al., 2004) realiza a comparação de uma análise que viola este princípio com outra que se enquadra neste princípio (cf. Figura 12). Como pode ser visto na Figura 12, em (A) tem-se uma derivação que viola este princípio metodológico, uma vez que utiliza do elemento zero (0) durante a análise de “*every girl*” e “*water*”. Isto significa que este elemento zero não foi declarado com superfície, violando este princípio. Já em (B), têm-se a análise da mesma frase, onde cada derivação combina o início da sentença com a próxima palavra em uma nova sentença de início. Com isto, este princípio não é violado e satisfaz as condições do princípio empírico.

No que se refere ao princípio empírico, (HAUSSER, 2006) define que a relação mais elementar entre as palavras em uma sentença é a ordem tempo-linear. Assim, a análise em tempo linear é baseada no princípio de possíveis continuações, o que se difere de outras análises que se baseiam em possíveis substituições (e.g., PSG⁴). Para atender a ideia de tempo linear, Hausser propôs um novo formalismo chamado de LAG. O LAG analisa palavra por palavra em uma sequência de tempo linear, combinando a sentença de início (i.e., palavras analisadas até o momento) com a próxima palavra baseando-se no conceito de possíveis continuações. Estas possíveis continuações são especificadas em regras que denotam padrões a serem seguidos durante a análise. Uma explicação mais detalhada do formalismo pode ser visto na Seção 2.4.1.

Para o princípio ontológico (i.e., *Internal*), a interpretação e produção de frases são analisadas como procedimentos cognitivos localizados no interior dos modos “*speaker*” e “*hearer*”. No que diz respeito ao princípio funcional (i.e., *Matching*), este se refere à linguagem no passado, presente ou futuro, objetos e eventos, relações e propriedades modelados em termos de correspondência de padrões entre o significado da linguagem e um contexto definido. Em outras palavras, o processo de interpretação e produção de

⁴ *Phrase Structure Grammar*

Figura 12 – Comparação entre as análises com base no princípio metodológico

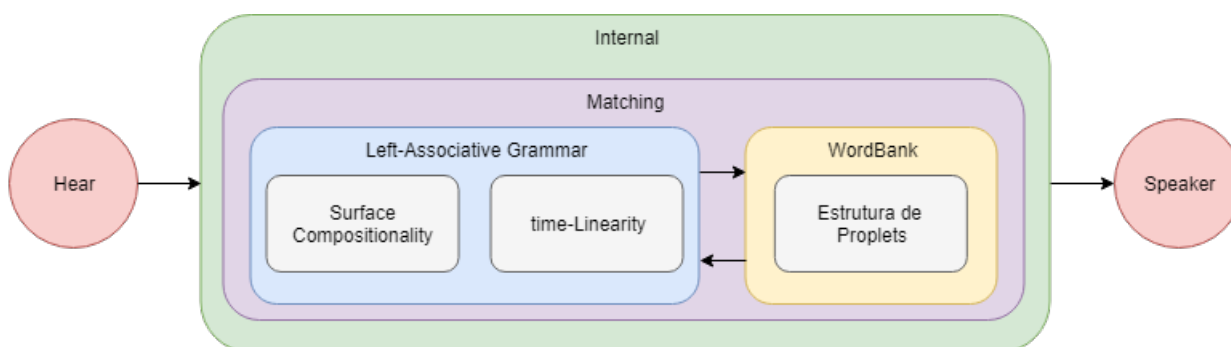


Fonte: do Autor. (Extraído de (HAUSSER, 2006))

frases deve ser realizado internamente em seus agentes cognitivos utilizando padrões de *Matching*. Estes padrões são abordados pelos princípios de *Surface Compositionality* e *time-Linearity*.

Para melhor compreender a teoria, na Figura 13 mostra-se como ela pode ser interpretada. Dado que a teoria faz alusão a uma comunicação entre duas pessoas, existe um modo “*hear*” que escuta o que o modo “*speak*” falar, tornando a comunicação mútua. Esta comunicação ocorre por meio de uma validação interna (i.e., *Internal Matching*) utilizando padrões gramaticais em uma estrutura definida em um *database*, denominado *WordBank*. Este *database* detém a representação do conhecimento em formato de *proplets*, o qual possui atributos atômicos que definem cada palavra existente em um domínio específico. Ambos os “modos” fazem uso da prática de *Internal Matching* usando o *Surface Compostionality* e o *time-Linearity* de base para fazer a comunicação acontecer.

Figura 13 – Ilustração da Teoria SLIM



Fonte: do Autor.

Como ferramentas que implementam esta teoria SLIM, o JSLIM é a mais recente⁵.

⁵ Existem outras ferramentas que implementam esta teoria: NEWCAT (HAUSSER, 1985) e JSLIM 1.0 (KYCIA, 2004). Entretanto, ambas estão defasadas e não possuem uma forma de acessar os detalhes da implementação. O JSLIM 2.0 está disponível em: www.linguistik.uni-erlangen.de/clue/en/research/jslim/download.html (Acessado em: 20-10-2017)

utilizando a linguagem de programação Java. Segundo (HANDL et al., 2009), esta ferramenta foi desenvolvida de modo a facilitar a codificação das gramáticas para morfologia, sintática e semântica. Vale destacar que o JSLIM apresenta um pré-processador de alo-morfos⁶, este refere-se ao método apresentado em (HAUSSER, 1989; HAUSSER, 2013) para realizar a flexão das palavras.

2.4.1 Left-Associative Grammar

O nome *Left-Associative Grammar* é motivado pela ordem de derivação de *left-associative*. A noção de *left-associative* é conhecida da lógica. Segundo (AHO; ULLMAN, 1977), quando combinamos os operadores para formar uma expressão, a ordem com que estes operadores são aplicados podem não ser óbvia. Por exemplo, $a + b + c$ pode ser interpretado como $((a + b) + c)$ ou como $(a + (b + c))$. Quando utilizado o símbolo “+” no *left-associative*, as operações são realizadas da esquerda para a direita, como em $((a + b) + c)$. O oposto (i.e., *right-associative*) ocorre em $(a + (b + c))$.

Neste sentido, a ordem de derivação do *left-associative* é linear no sentido de que regularmente adiciona uma palavra após a próxima, sendo também linear ao tempo, uma vez que a direção do crescimento corresponde à direção do tempo. Assim, esta gramática se baseia na ideia de possíveis continuções, isto é, começando com a primeira palavra da sentença, a gramática descreve as possíveis continuções para cada início de sentença, especificando regras que podem realizar a próxima composição gramatical (i.e., adicionar a próxima palavra). A ordem de derivação tempo-linear e a caracterização estrutural de possíveis continuções é formalizada no esquema de regras específicas da LAG (cf. Figura 14).

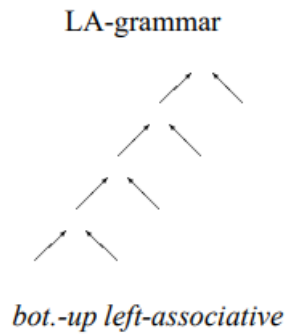
Figura 14 – Exemplo de regra gramatical LAG

$$r_i: cat_1 \ cat_2 \Rightarrow cat_3 \ rpi_i$$

Fonte: (HAUSSER, 2013).

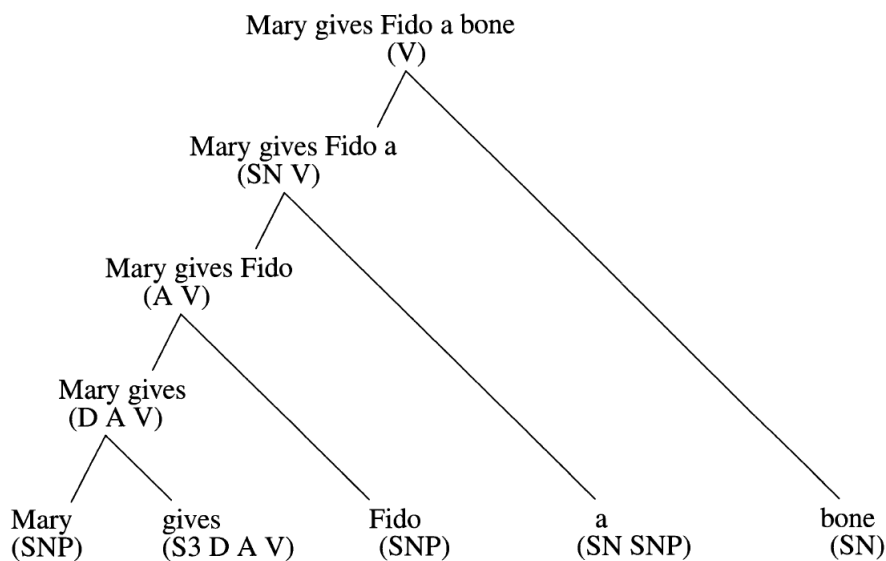
Conforme apresentado na Figura 14, a regra consiste do nome r_i , os padrões de categoria (cat_1 , cat_2 e cat_3) e o pacote de regras (i.e., rpi_i). O padrão de categorias define a operação que mapeia a sentença de início (i.e., ss) (cat_1) e a próxima palavra (i.e., nw) (cat_2) em uma nova sentença de início (i.e., ss') (cat_3). O resultado positivo da aplicação desta regra r_i é o estado definido como um par ordenado (i.e., $ss' \ rpi_i$). Na próxima combinação, as regras do pacote de regras rpi_i são aplicadas ao ss' e uma nova palavra nw . Uma vez que a análise da LAG começa com as palavras iniciais de entrada, sua ordem de derivação conceitual é *bottom-up left-associative* (HAUSSER, 1992) (cf. Figura 15).

⁶ Variação de um morfema sem mudança no seu significado.

Figura 15 – *Bottom-up Left-Associative*

Fonte: (HAUSSER, 2013).

No que se refere à aplicação na linguagem natural, na Figura 16 mostra-se um exemplo de análise utilizando o LAG para a frase “*Mary gives Fido a bone*”. Neste exemplo, a análise começa em “*Mary*”, que tem a categoria *SNP* (i.e., frase nominal singular) que é combinado com a categoria *S3* (i.e., singular na terceira pessoa) existente na próxima palavra “*gives*”. Assim, o *S3* é cancelado de modo que sobrem apenas as categorias (*D*, *A*, *V*). Ao receber a próxima palavra “*Fido*” (definida por *SNP*), a categoria *D* (i.e., dativo) presente no verbo “*gives*” é cancelada, sobrando somente as categorias *A* (i.e., acusativo) e *V* (i.e., verbo). Em seguida, a próxima palavra “*a*” tem como categoria (*SN SNP*), neste sentido, cancela-se *A* existente em “*gives*” substituindo por *SN* (i.e., substantivo singular) definido em “*a*”. Por fim, a palavra “*bone*” é representada pela categoria *SN*, a qual está presente na definição de “*gives*” e que é cancelada de modo que sobre somente a categoria *V* que define o verbo “*gives*”.

Figura 16 – Exemplo da análise da frase “*Mary gives Fido a bone*” utilizando a LAG

Fonte: (HAUSSER, 2013).

2.4.2 DB Semantics

O modelo DBS consiste em estruturas de dados para o processamento da linguagem e a representação do conhecimento do contexto, juntamente com três gramáticas LAG: uma gramática para a análise da linguagem, uma gramática paralela para geração de linguagem natural e uma terceira gramática que fornece o mecanismo de inferência em todo o conhecimento de contexto formalmente representado. No que se refere as estruturas de dados, estas são chamadas de *proplets*⁷ e são definidas como estruturas planas (atômicas) constituídas por pares de característica-atributo. Por se referenciar às proposições básicas, os *proplets* fazem uso da estrutura *functor-argument*. Neste sentido, (HAUSSER, 1999) destaca que os atributos principais desta estrutura são definidos por *functor*, *argument* e *modifier*. Diante disto, uma proposição consiste de um *functor* (e.g., verbo), que combina com um número de *argument* (e.g. substantivo), tendo o *modifier* (e.g., adjetivos e advérbios) um opcional que pode ser aplicado tanto ao *functor* quanto ao *argument*. Um exemplo desta estrutura é apresentado na Figura 17.

Conforme mostrado na Figura 17, a definição da estrutura divide-se em *types* e *proplets* (também chamados de *proplets tokens*). Os *proplets* de tipo são como dados base (i.e., *owner proplet* e os *proplets* são os dados de relacionamento (i.e., *member proplet*). Isto remete à uma conexão de banco de dados clássica, sendo assim, o *Word Bank* é modelado como um banco de dados relacional. Os relacionamentos ocorrem com base nos atributos de cada *proplet*, ou seja, o *proplet* de tipo com o atributo “*verb*” tem o valor “*contain*”, este, por sua vez, possui como membros os relacionamentos aos *proplets* com os valores “*field*” e “*square*”. Com isso, têm-se a definição de uma relação (1:n) entre os dados principais (i.e., *type*) e os dados membros (i.e., *proplet*) dentro da base de dados, chamado de *Word Bank* (HAUSSER, 2013). Esta relação é definida pelo atributo *prn* (i.e., número da proposição). No caso dos atributos *arg*, *fnc* e *mdd/mdr*, estes são específicos do *functor*, *argument* e *modifier*, respectivamente. Para o *proplet argument*, este terá um atributo *fnc* com valores referentes ao *functor*, ou seja, a qual verbo este substantivo está relacionado baseado em determinado *prn*. O mesmo ocorre com o *functor* (utiliza *arg*) e *modifier* (utiliza *mdd*).

Para a análise e geração de linguagem natural no DBS, utiliza-se de três agentes que usam da LAG, chamadas de LA-Hear, LA-Think e LA-Speak, mencionado na Seção 2.4. O LA-Hear recebe o *input* e interpreta os sinais de modo a validá-los sintaticamente e semanticamente, armazenando, posteriormente, estas informações no *Word Bank* em formato de *proplet*. O objetivo do LA-Think é de selecionar os conteúdos armazenados de modo a servir de *input* para o LA-Speak, que por sua vez reordena o conteúdo em linguagem natural. Diante disto, (HAUSSER, 2001b) explica que LA-Think tem a função de identificar “*what to say*” (i.e., o que dizer) e o LA-Speak tem a função de identificar

⁷ Este termo faz analogia ao *droplet* que se refere a parte básica de uma proposição elementar. Uma proposição elementar consiste de um *functor*, que combina com um número característico de *arguments*. *Modifiers* são opcionais e podem ser aplicados a *functors* assim como *arguments*.

Figura 17 – Exemplo de *proplet* e seus membros

TYPES	PROPLETS	
$\left[\begin{array}{l} \text{func: } \textit{contain} \\ \text{ARG:} \\ \text{cnj:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{func: } \textit{contain} \\ \text{ARG:field triangle} \\ \text{cnj: 23 and 24} \\ \text{prn: 23} \end{array} \right]$	$\left[\begin{array}{l} \text{func: } \textit{contain} \\ \text{ARG:field square} \\ \text{cnj: 23 and 24} \\ \text{prn: 24} \end{array} \right]$
$\left[\begin{array}{l} \text{arg: } \textit{field} \\ \text{FUNC:} \\ \text{id:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{arg: } \textit{field} \\ \text{FUNC: contain} \\ \text{id: 7} \\ \text{prn: 23} \end{array} \right]$	$\left[\begin{array}{l} \text{arg: } \textit{field} \\ \text{FUNC: contain} \\ \text{id: 7} \\ \text{prn: 24} \end{array} \right]$
$\left[\begin{array}{l} \text{arg: } \textit{square} \\ \text{FUNC:} \\ \text{id:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{arg: } \textit{square} \\ \text{FUNC: contain} \\ \text{id: 9} \\ \text{prn: 24} \end{array} \right]$	
$\left[\begin{array}{l} \text{arg: } \textit{triangle} \\ \text{FUNC:} \\ \text{id:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{arg: } \textit{triangle} \\ \text{FUNC: contain} \\ \text{id: 8} \\ \text{prn: 23} \end{array} \right]$	

Fonte: (HAUSSER, 2001a)

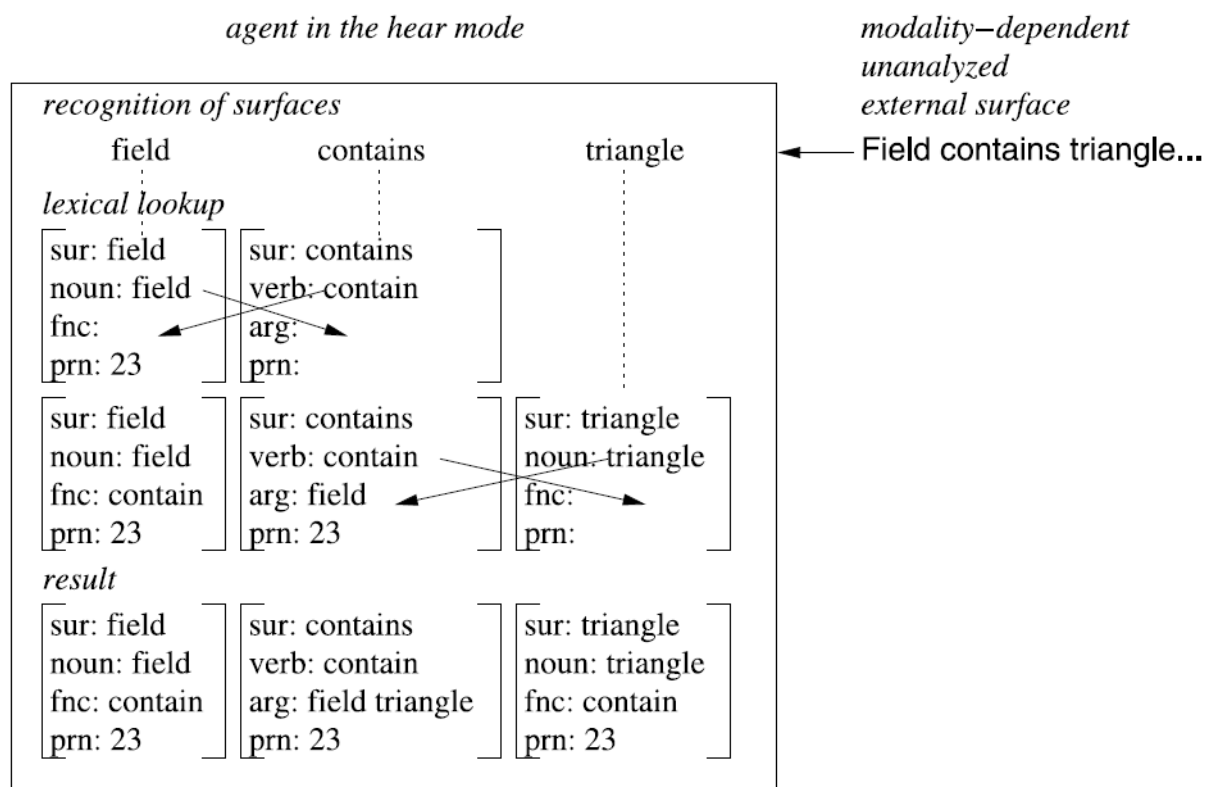
“*how to say it*” (i.e., como dizer isso). O componente que intermedia todo o funcionamento do DBS é chamado de LA-Motor.

Na Figura 18 mostra-se a derivação do LA-Hear, nela, observa-se que é feito o mapeamento das palavras de modo a reconhecer a estrutura de *proplet* adequada de cada palavra. Este reconhecimento é feito de forma incremental e linear seguindo as regras da LAG (cf. Seção 2.4.1), a qual procura pela próxima palavra de modo a concatenar com a sentença inicial transformando em uma nova sentença de início. Esta sequência continua até a derivação não ter mais palavra para analisar ou encontrar alguma outra que não é reconhecida pelo sistema.

No que tange ao LA-Think, este realiza a navegação no conteúdo do *Word Bank*. Um exemplo desta navegação é apresentada na Figura 19. Utilizando o valor de continuação “*yesterday*” e o valor do *prn* do *proplet* “*buy*” como endereço, a etapa 1 começa do *buy* para o *yesterday*. Na etapa 2_3, a navegação parte de *yesterday* para *buy* e de *buy* para *Mary*. Na sequência, a etapa 4 navega de *Mary* para *buy* para na etapa 5 ir de *buy* para *book*. A partir do *book*, a etapa 6 navega para *cookie* que por fim retorna à *book* e em seguida à *buy*.

Em relação ao LA-Speak, este tem o objetivo de produzir linguagem natural. Na Figura 20 mostra-se um exemplo desta derivação. Esta derivação se inicia obtendo o valor do atributo *prn* do *proplet* *field*, neste caso é o 23. Com base no atributo *fnc* do *proplet* *field*, identifica-se que a continuação é a palavra *contains*. Em seguida, identifica o *owner proplet* de modo a buscar o *prn* 23 definido anteriormente pelo *field*. Neste sentido, o *member proplet* identificado possui o atributo *arg* com os valores *field* e *triangle*. O

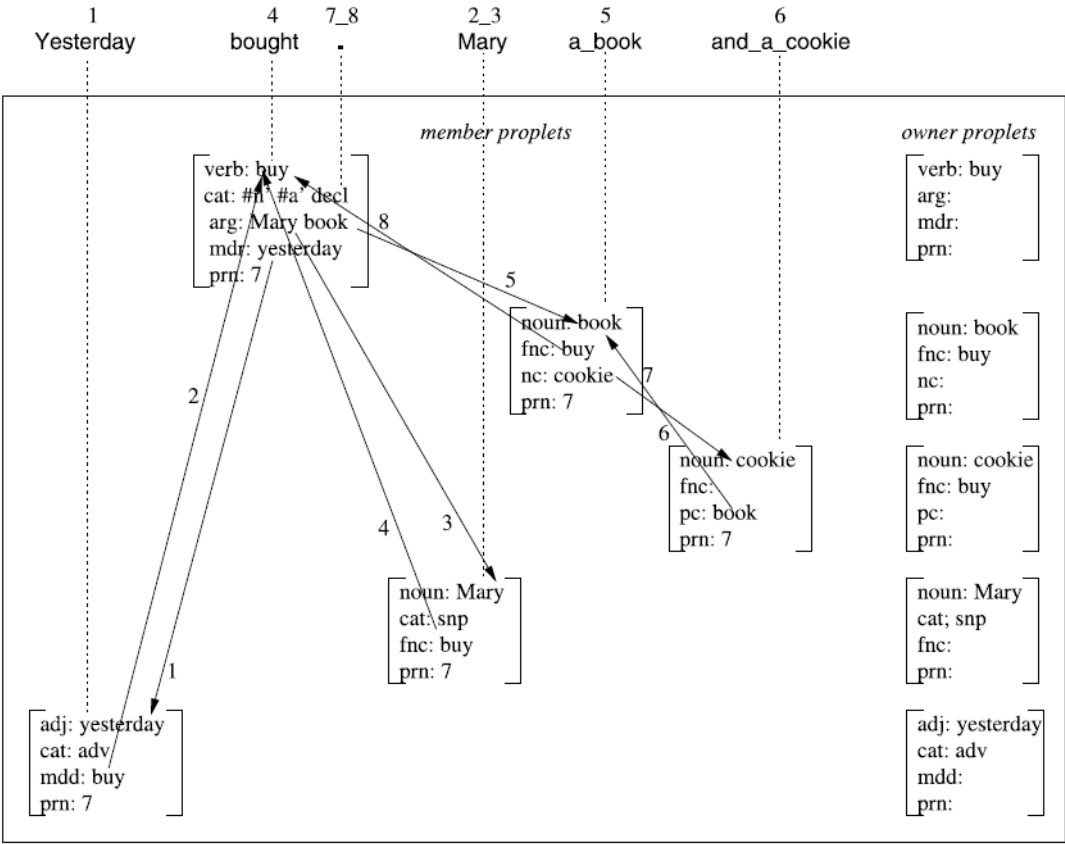
Figura 18 – Exemplo da derivação LA-Hear



Fonte: (HAUSSER, 2001a)

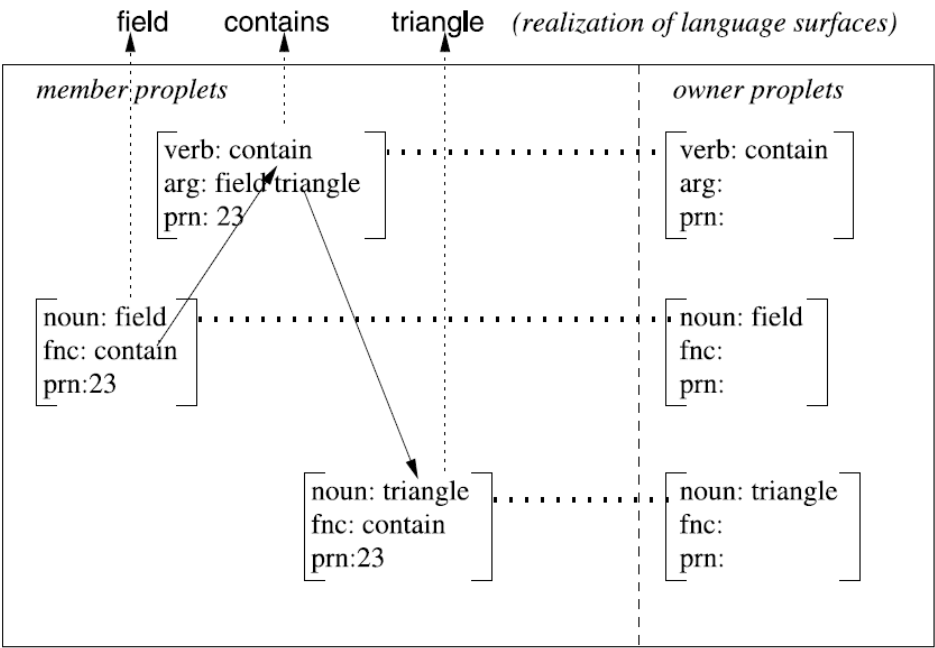
primeiro valor, *field*, confirma a relação de *contain* com *field*, já o segundo valor *triangle* é o novo próximo *proplet*. Assim, identifica-se novamente o *owner proplet* que faz referência ao *triangle*, de modo a encontrar o *member proplet* com o valor do atributo *prn*, sendo ele o número 23.

Figura 19 – Exemplo da navegação LA-Think



Fonte: (HAUSSER, 2013)

Figura 20 – Exemplo de derivação LA-Speak



Fonte: (HAUSSER, 2013)

3 TRABALHOS RELACIONADOS

Neste capítulo apresenta-se uma discussão sistematizada dos trabalhos relacionados a esta proposta. O restante deste capítulo está estruturado da seguinte forma: na Seção 3.1 apresentam-se os critérios utilizados para a avaliação dos trabalhos relacionados; na Seção 3.2 os trabalhos relacionados são discutidos e por fim na Seção 3.3 realiza-se uma análise comparativa entre os trabalhos.

3.1 CRITÉRIOS PARA A AVALIAÇÃO DOS TRABALHOS

Os trabalhos apresentados neste capítulo são avaliados e comparados de acordo os seguintes critérios:

- **C1 – Léxico Personalizado.** Indica se a solução faz uso de um léxico personalizado do usuário. Isto possibilita que o usuário se comunique utilizando um léxico com conceitos de seu conhecimento;
- **C2 – Gramática Visual.** Indica se a solução faz uso de algum guia que facilite a composição de uma frase (e.g., Chave de Fitzgerald e CS). Isto possibilita ao usuário com limitação intelectual, motora ou na fala, organize com maior facilidade a disposição correta das palavras que compõem uma frase;
- **C3 – Sugestão de Candidatos.** Indica se a solução utiliza algum método para a sugestão de pictogramas candidatos. Com a sugestão de candidatos, apenas os pictogramas que tenham concordância sintática e semântica são apresentados, o que reduz a quantidade de cliques para acessar um pictograma de interesse;
- **C4 – Implementação de Ferramenta.** Indica se a solução possui uma implementação disponível para utilização. A implementação da ferramenta mostra que a solução é factível e oferece a possibilidade de avaliá-la;
- **C5 – Utilização de Word Bank.** Indica se a solução faz uso de alguma estrutura de armazenamento de dados utilizando informações referentes ao léxico e ao relacionamento com outras palavras do léxico. Esta prática permite que as etapas de análise sintática e semântica tenha uma redução no tempo de resposta, melhorando a performance e reduzindo a frustração do usuário.

3.2 APRESENTAÇÃO DOS TRABALHOS

Nesta seção apresentam-se os principais trabalhos relacionados a esta proposta, a saber: *Pictogrammar - an AAC device based on a semantic grammar* (MARTÍNEZ-SANTIAGO et

al., 2016), *Using Semantic Authoring for Blissymbols Communication Boards* (NETZER; ELHADAD, 2006a) e *Compansion system for pictogram-based AAC application in Catalan* (PAHISA-SOLÉ, 2012). Estes trabalhos foram selecionados a partir de uma busca pela *string* “*Semantic Parser*” AAC em bibliotecas digitais (i.e., Google Scholar, IEEEExplore), bem como na verificação dos trabalhos relacionados dos mesmos.

MARTÍNEZ-SANTIAGO et al.

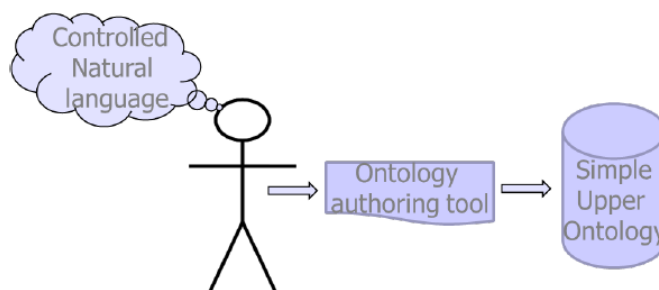
O trabalho proposto por (MARTÍNEZ-SANTIAGO et al., 2016) descreve um SCAA para comunicadores iniciantes. Os autores definem este SCAA a partir de três componentes (cf. Figura 21): linguagem natural controlada, ontologia para modelar a linguagem em nível conceitual e ferramenta de criação. Segundo os autores a linguagem natural controlada corresponde a comunicação simbólica a ser aprendida e/ou utilizada pelo usuário. Em (MARTÍNEZ-SANTIAGO et al., 2015), a linguagem controlada é definida como qualquer segmento de linguagem natural especificado com um conjunto formal de regras. Para o segundo componente, os autores fazem uso de uma ontologia denominada *Simple Upper Ontology* (SUPO), a qual corresponde a um conhecimento básico e geral do mundo. Este conhecimento é enriquecido por informações sintáticas e semânticas que possibilitam a interação com o mundo real. O terceiro componente, a ferramenta de criação *Pictogrammar*, faz a integração entre o usuário e a SUPO. Os dados de entrada do usuário são informações contidas na ontologia *PictOntology*, a qual oferece como vocabulário um conjunto de pictogramas mapeados em (1:1) ao vocabulário da SUPO.

A linguagem natural controlada apresentada no trabalho de Martínez-Santiago é composta por 621 palavras que incluem conceitos concretos bem como suas propriedades (e.g., cores, formas, tamanho, temperatura, sentimentos e emoções), eventos, preposições, interjeições e pronomes. Com base nesse vocabulário, o autor destaca que é possível expandi-lo com regras morfossintáticas de gênero e número, descartando regras mais complexas como o tempo verbal. Esta linguagem natural é representada pela SUPO, cujo objetivo é o desenvolvimento de uma gramática semântica com a cobertura da linguagem natural do usuário. Para o modelo semântico, Martínez-Santiago adapta o FrameNet (BAKER; FILLMORE; LOWE, 1998) aproveitando parte de suas taxonomias e estruturas de argumento-predicado no nível semântico para cada verbo. O modelo sintático é implementado utilizando o *Grammatical Framework* (GF) (RANTA, 2011) e sua biblioteca de recursos com modelos morfossintáticos e sintáticos. Diante disso, os autores converteram o conhecimento semântico contido na SUPO para as classes abstratas do GF. Para o sintático, utilizaram as classes concretas e a biblioteca de recursos. Estas estruturas são consumidas posteriormente pelo preditor semântico desenvolvido por (ANGELOV,), o qual é utilizado para restringir e oferecer pictogramas candidatos ao usuário.

Para a ferramenta de criação, o *Pictogrammar*, (MARTÍNEZ-SANTIAGO et al., 2016) atribui a tarefa de utilizar o resultado da união do modelo semântico da SUPO com o

modelo morfossintático e sintático do GF, usando como suporte de entrada de dados os pictogramas disponíveis no *PictOntology*. Com a combinação destas tecnologias o autor apresenta algumas possíveis melhorias na comunicação, entre elas está a redução da quantidade de pictogramas disponíveis durante a montagem da expressão. Para isso, oferece na tela principal apenas pictogramas que tenham relação semântica com os selecionados.

Figura 21 – Diagrama Simplificado da Proposta



Fonte: (MARTÍNEZ-SANTIAGO et al., 2016)

Em resumo, o trabalho apresenta uma linguagem controlada que, segundo os autores, pode ser compartilhada e personalizada, criando um léxico personalizado ao usuário com conceitos de seu conhecimento. Por se basear em uma SCAA para crianças em idade pré-escolar com o objetivo desenvolver suas habilidades sintáticas e semânticas, não há indícios de uma gramática visual que aprimore a capacidade de compreensão dos pictogramas, nem guie, de forma visual, a sequência a ser seguida na estruturação da frase. Por dispor de uma técnica de predição semântica de pictogramas, este trabalho oferece uma maneira de delimitar o conjunto de pictogramas disponíveis durante a construção da sentença, sugerindo somente pictogramas relacionados sintática e semanticamente. No que se refere a implementação de ferramenta, os autores apresentam a interface de um protótipo (cf. Figura 22), entretanto, estes não mostram informações referentes a sua implementação tampouco a forma de acessar a solução. Por fim, não há evidência de utilização de um *Word Bank* com o intuito de reduzir o tempo de resposta do sistema, somente a consulta a ontologia para fins de validação semântica. Na Tabela 2, de acordo com a presença (+) e ausência (-) dos critérios elencados na Seção 3.1, apresenta-se uma análise consolidada do trabalho em questão.

NETZER; ELHADAD

O trabalho desenvolvido por (NETZER; ELHADAD, 2006a) apresenta uma ferramenta de criação que gera linguagem natural em Inglês e Hebraico utilizando Blissymbols¹. Esta ferramenta visa encorajar o usuário a criar mensagens na ordem correta. Sua estratégia

¹ Linguagem gráfica referente ao significado, criada por Charles Bliss (BLISS, 1965) para ser usada como uma linguagem de escrita universal, posteriormente utilizado em aplicações de CAA.

Figura 22 – Interface do Pictogrammar



Fonte: (MARTÍNEZ-SANTIAGO et al., 2016)

Tabela 2 – Critérios abrangidos pelo trabalho de MARTÍNEZ-SANTIAGO et al.

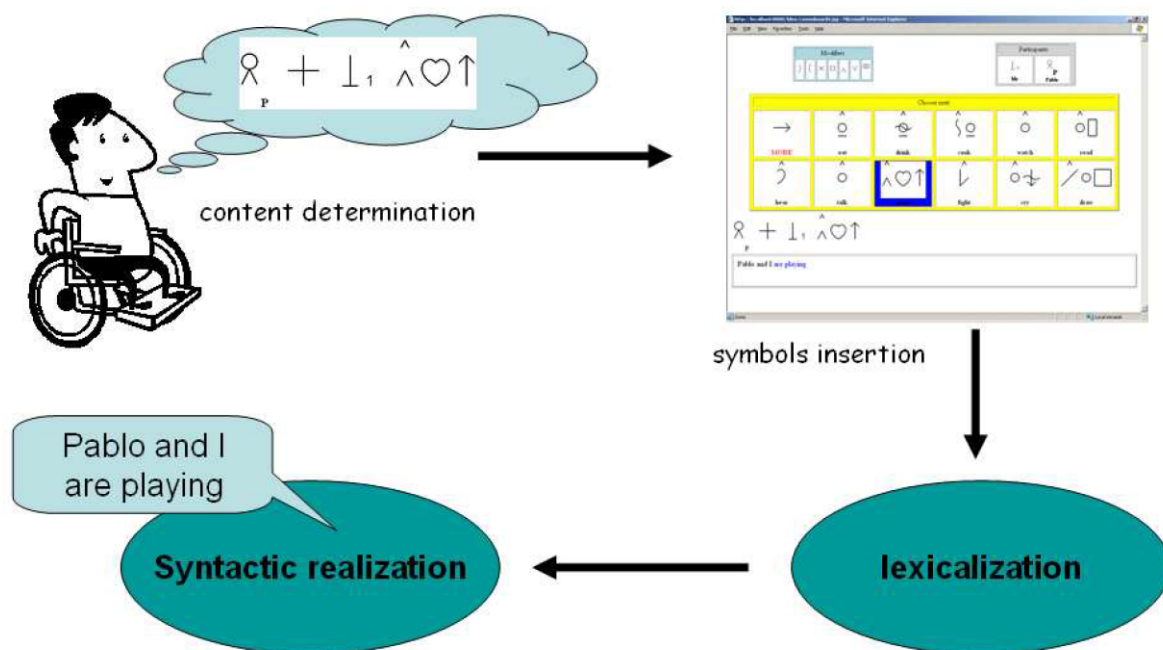
CRITÉRIO	ENGLOBA
C1 - Léxico Personalizado	+
C2 - Gramática Visual	-
C3 - Sugestão de Candidatos	+
C4 - Implementação de Ferramenta	-
C5 - Utilização de Word Bank	-

Fonte: do Autor.

combina uma base de dados léxico para a criação de uma ontologia, e uma ferramenta de criação semântica com realizador sintático² de modo a obter símbolos sintática e semanticamente relacionados ao símbolo selecionado anteriormente. Por exemplo, dado um símbolo que denota um verbo que restringe seu complemento em algo comestível, apenas símbolos que se encaixam nessa restrição serão apresentados ao usuário. Neste sentido, o usuário cria sua sentença sintaticamente e semanticamente bem estruturada em forma de linguagem telegráfica que em seguida é reescrita pela ferramenta de criação em linguagem natural. Na Figura 23 apresenta-se o fluxo de informação da proposta.

Na Figura 24 apresenta-se a arquitetura do sistema, a qual é dividida em três módulos: ontologia, seletor léxico e realizador sintático. Nesta figura também são mostradas as estruturas de dados e as fontes de conhecimento utilizadas pelo sistema. Para a criação do léxico, o autor utiliza de dois recursos: Léxico Bliss e Léxico de Verbos Integrados. Para

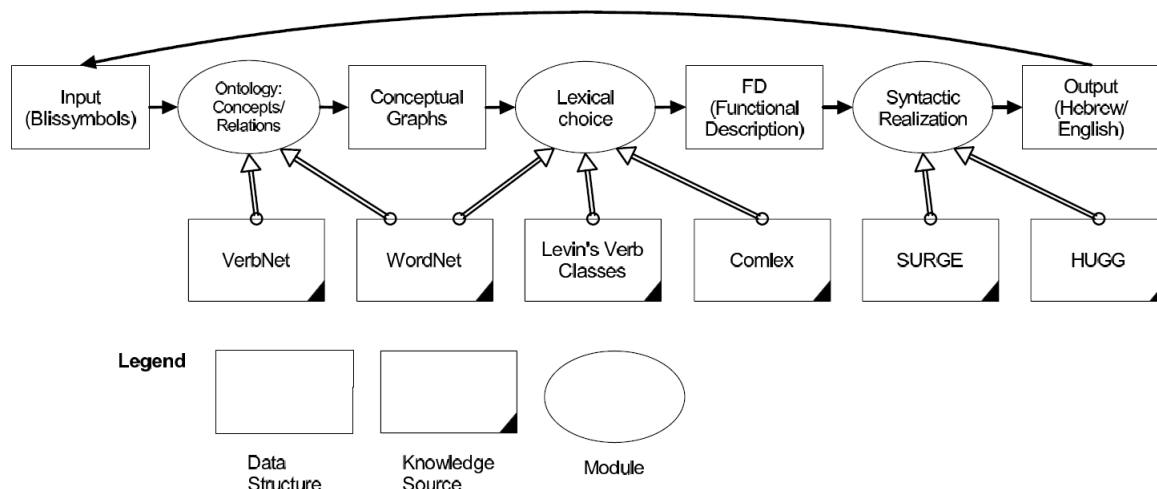
² Subtarefa da geração de linguagem natural que envolve a criação de um texto em linguagem humana (i.e., Inglês, Português, etc.) a partir de uma representação sintática.

Figura 23 – Fluxo de informação do sistema de criação do *Blissymbols*

Fonte: (NETZER, 2006b)

o primeiro, foi criado e implementado os *Blissymbols* para Inglês e Hebraico que juntos totalizam 2200 símbolos disponíveis no sistema, este léxico também provê uma ferramenta *web* para a criação, edição e exclusão dos símbolos e suas palavras. Na definição do segundo recurso, (NETZER, 2006b) combinou a estrutura e a informação semântica de diversas fontes léxicas (i.e., WordNet (MILLER, 1995), VerbNet (KIPPER et al., 2000), *English Verb Classes and Alternations* (EVCA) (LEVIN, 1993) e o COMLEX *Syntax Dictionary* (COMLEX) (GRISHMAN; STERLING, 1989) em um léxico com verbos em Inglês. Este léxico foi formatado como uma extensão ao *Systemic Unification Realization Grammar of English* (SURGE), o qual é um realizador sintático. Com uma gramática reutilizável, o SURGE provê uma linguagem de especificação composicional de entrada e padrões que determinam as funções das palavras baseadas nas descrições funcionais, ordem dos constituintes, além de realizar processamento morfológico e a pronominalização sintática (i.e., substituição de um nome ou de um sintagma nominal por um pronome).

A ontologia é definida como o núcleo do processo de criação semântica. Ela provê a estrutura hierárquica dos conceitos, suas relações e propriedades – neste caso os verbos, substantivos, adjetivos e advérbios representados pelos *Blissymbols* –, além de informações como sinônimos, antecessores na hierarquia e relações obrigatórias dos conceitos, se estas existirem. Para cada palavra do Léxico Bliss, o autor escolheu manualmente um *synset* (i.e., conjunto do *WordNet* que denota o significado de um conjunto de palavras sinônimas) de acordo com o WordNet.

Figura 24 – Arquitetura do Sistema de criação *Blissymbols*

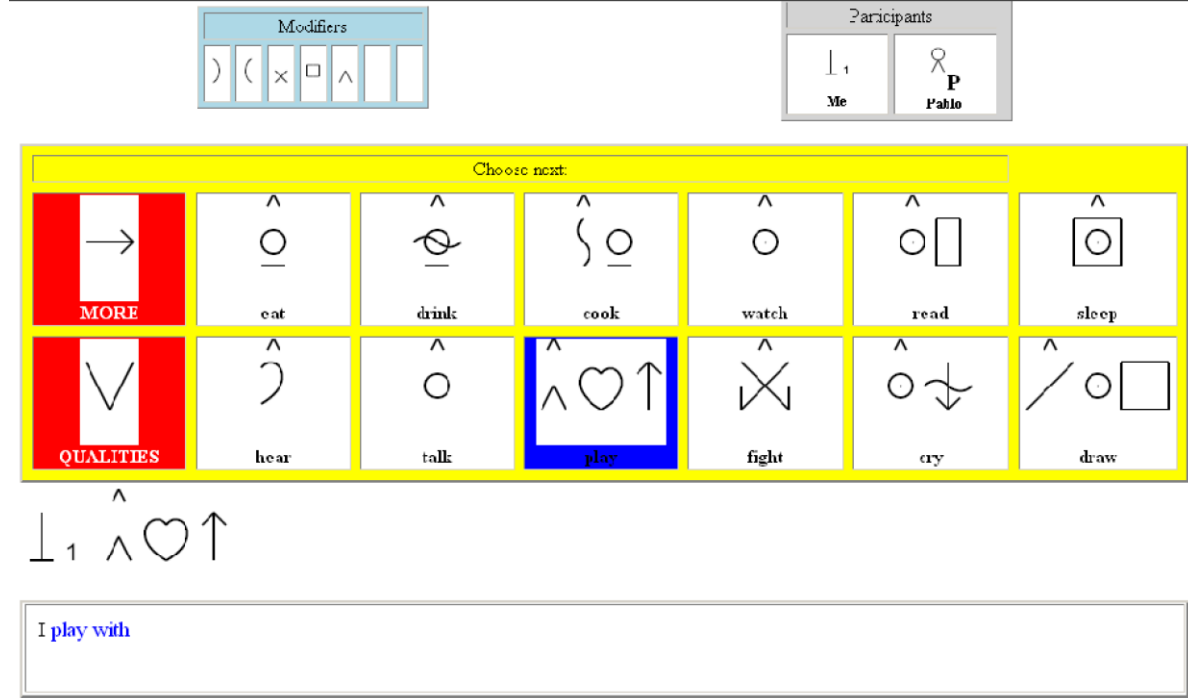
Fonte: (NETZER, 2006b)

Os dois últimos módulos que compõem o sistema são chamados de seletor lexical (traduzido do inglês "*lexical chooser*") e realizador sintático. Para o seletor lexical existe o mapeamento dos símbolos em forma de grafos conceituais associados a um léxico de entrada com a respectiva Descrição Funcional (DF)³. O objetivo deste módulo é de converter o símbolo e o léxico em uma interface DF de entrada ao realizador sintático SURGE. Por fim, após a realização sintática, a frase gerada é apresentada ao usuário em forma de linguagem natural.

Em resumo, o trabalho apresenta uma linguagem controlada que pode ser personalizada por uma ferramenta de criação, permitindo que o léxico seja específico para cada usuário. Não há indícios de uma gramática visual que facilite a compreensão dos símbolos, nem guie, de forma visual, a sequência a ser seguida na construção de uma frase. Por dispor de uma ontologia com informações semânticas, a cada etapa na construção da frase, o sistema restringe o número de possíveis símbolos com base na relação sintática e semântica dos mesmos. No que tange à implementação da ferramenta, apresenta-se na Figura 25 a interface que mostra o estado de funcionamento durante uma restrição semântica, entretanto, não há uma forma de acesso à ferramenta. No caso do critério C5 (utilização de *Word Bank*), o autor utiliza somente a Ontologia para fazer a restrição semântica, não utilizando um *Word Bank* com o intuito de reduzir o tempo de resposta do sistema. Na Tabela 3 apresenta-se uma análise consolidada do trabalho em questão, de acordo com a presença (+) e ausência (-) dos critérios elencados na Seção 3.1.

³ Todo o conhecimento linguístico é representado como um conjunto de *features*. Estas *features* possuem informações linguísticas do constituinte com atributos funcionais que marcam sua funcionalidade no contexto.

Figura 25 – Interface da proposta de Netzer



Fonte: (NETZER, 2006b)

Tabela 3 – Critérios abrangidos pelo trabalho de NETZER

CRITÉRIO	ENGLOBA
C1 - Léxico Personalizado	+
C2 - Gramática Visual	-
C3 - Sugestão de Candidatos	+
C4 - Implementação de Ferramenta	-
C5 - Utilização de Word Bank	-

Fonte: do Autor.

PAHISA-SOLé

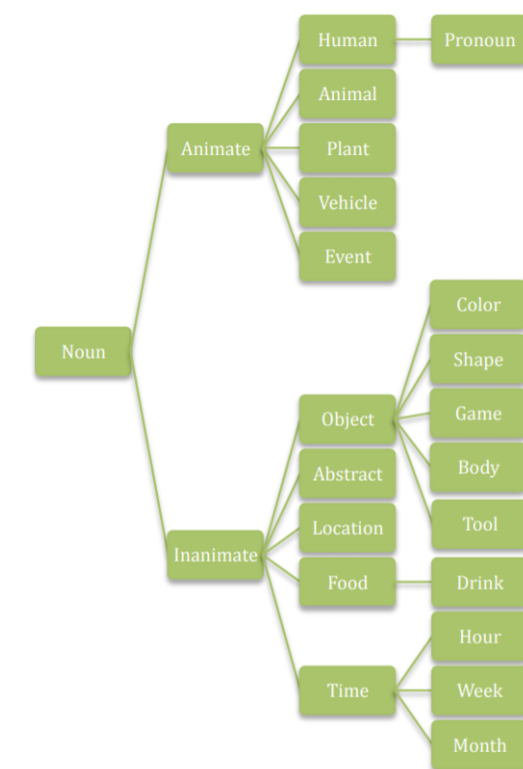
O trabalho de (PAHISA-SOLé, 2012) apresenta um COMPANSION que expande frases em linguagem telegráfica em frases em linguagem natural (neste caso Catalão), no contexto de uma aplicação em CAA. O sistema foi projetado para melhorar a comunicação dos utilizadores, entretanto, também pode ser utilizado para auxiliar na alfabetização e reabilitação da linguagem. O COMPANSION desenvolvido é composto por dois componentes principais: um *parser* sintático-semântico para validação sintática e semântica dos pictogramas e um gerador para a construção da frase de saída. Como dados de entrada, o sistema proposto por PAHISA-SOLé utiliza um vocabulário controlado enriquecido de informações sintáticas e semânticas. A partir deste vocabulário, o *parser* extrai os padrões

frasais e informações associadas as palavras e seus modificadores. Por fim, o sistema gera a frase de saída a partir da árvore de análise resultante, reordenando e flexionando as palavras, acrescentando artigos, preposições e outros elementos necessários para que a expressão esteja corretamente estruturada em língua natural.

Para o léxico de entrada, o sistema em questão utilizou um vocabulário de 600 palavras provenientes do vocabulário CACE⁴, as quais foram enriquecidas com informações sintáticas e semânticas por meio de anotação manual em estrutura hierárquica (cf. Figura 26). O autor armazena essa estrutura em um banco de dados.

Devido à limitação de tempo, o autor criou um protótipo (cf. Figura 27) para entrada de dados que trabalha somente com as palavras, porém, estas se referem a pictogramas. Observa-se que o sistema, mesmo sendo um protótipo sem pictogramas, dispõe de uma organização por categoria sintática acompanhado de um guia de cores. Para interagir com o mesmo, basta selecionar as palavras e clicar na opção definida pelo símbolo "+" e por fim clicar em um botão de geração. Destaca-se que o protótipo espera que o usuário tente preencher a estrutura de forma correta.

Figura 26 – Hierarquia Anotada dos Substantivos



Fonte: (PAHISA-SOLé, 2012)

Seguindo a abordagem COMPANSION proposta por (PENNINGTON; MCCOY, 1998) o núcleo do sistema é um *parser* semântico que interpreta os dados de entrada fazendo uso

⁴ Vocabulário em Catalão para Comunicação Suplementar e Alternativa. UTAC-CACE: <<http://www.utac.cat/noticies/utac-cacejadisponibleperadescarregarenquatreversions>>

Figura 27 – Interface protótipo COMPANSION

The interface is divided into several color-coded sections for selecting words:

- Noms (Plural, Femení):**
 - Humans:** Pronoms: això; Persones: actor.
 - Objectes o éssers animats:** Animals: abella; Vehicles i Altres: Barça; Plantes: arbre.
- Llocs i Hores:** Llocs: Amèrica; Hores: cinc.
- Temps:** General: any; Setmana: dijous; Mesos: abril.
- Verbs:** Verbs més comuns: Verb: agafar.
- Adjectius i Adverbis:**
 - Adjectius:** Per tot: alt; Animat: alegre; Objecte: arrissat.

Fonte: (PAHISA-SOLé, 2012)

de *frames* ou *slots*. Esses *slots* são estruturas conceituais que representam o significado de uma frase, descrevendo os casos ou papéis semânticos que cada um dos constituintes da frase tem em relação aos outros. Quando o sistema recebe a entrada, o *parser* semântico designa um verbo como o componente principal da expressão, todas as outras palavras da frase são utilizadas para preencher os papéis semânticos anotados que correspondem ao verbo principal.

O componente gerador tem a responsabilidade de inserir o conteúdo faltante na frase telegráfica para que fique em linguagem natural. Ele é composto por quatro módulos independentes, a saber:

1. **Ordenador de *slot*:** reordena os componentes da frase de acordo com o tipo de frase;
2. **Ordenador de palavra, preposição e concordância:** reordena internamente as palavras em cada uma das diferentes posições estruturais, este módulo acrescenta ainda as preposições e garante que as palavras possuam concordância de gênero e número;
3. **Tratamento de artigo:** contrai os artigos com os substantivos quando necessário;
4. **Conjugador de verbos:** realiza a modificação do verbo conforme seu sujeito e tempo verbal;
5. **Limpeza:** trata dos detalhes finais e obtém a frase pronta para a saída.

Em resumo, o trabalho apresenta uma linguagem controlada que é pré-definida e anotada manualmente, não disponibilizando uma forma de gerar automaticamente um léxico personalizado para utilização no sistema COMPANSION. O autor define um protótipo que organiza as informações por meio de uma chave de cores, que facilita a identificação

das categorias sintáticas e guia o usuário. De acordo com o autor, não há uma forma de sugerir candidatos semânticos por acreditar que o usuário o faça numa ordem correta. No sentido da implementação da ferramenta, o autor apresenta uma figura do protótipo o qual trabalha com palavras referenciando pictogramas, entretanto não há uma forma de obter acesso a mesma. Por ter feito a anotação manual do vocabulário o autor optou por salvar estas informações em um *Word Bank* que armazena as informações semânticas em uma estrutura de *features* que ele denomina de *slot*. Na Tabela 4, de acordo com a presença (+) e ausência (-) dos critérios elencados na Seção 3.1, apresenta-se uma análise consolidada do trabalho em questão.

Tabela 4 – Critérios abrangidos pelo trabalho de PAHISA-SOLé

CRITÉRIO	ENGLOBA
C1 - Léxico Personalizado	-
C2 - Gramática Visual	+
C3 - Sugestão de Candidatos	-
C4 - Implementação de Ferramenta	-
C5 - Utilização de Word Bank	+

Fonte: do Autor.

3.3 ANÁLISE COMPARATIVA

Os trabalhos relacionados a esta proposta foram analisados de acordo com a presença (+) e ausência (-) dos critérios elencados na Seção 3.1. Na Tabela 5 apresenta-se a análise comparativa desses trabalhos. Para o critério C1 (Léxico Personalizado), apenas o trabalho de PAHISA-SOLé não propõe uma forma de personalizar o léxico com novas palavras por possuir um vocabulário que foi manualmente anotado, o que impossibilita a personalização (i.e., criação, edição e remoção de novos elementos do vocabulário de maneira automatizada). Isto limita o poder de expressão do usuário, uma vez que ele tem disponível apenas o léxico anotado manualmente. Por conta disto, o usuário não consegue manter uma evolução no aprendizado da língua. No critério C2 (Gramática Visual), somente o trabalho de PAHISA-SOLé demonstra ter uma organização visual baseada em cores que facilitam na criação de uma frase pelo usuário. Os demais trabalhos não apresentam evidências de que possuam este tipo de organização e guia visual. A ausência deste critério dificulta a comunicação do usuário, dado que o mesmo não cria associações visuais das cores com os pictogramas e nem facilita no ordenamento para uma frase bem formada. Com relação ao critério C3 (Sugestão de Candidatos), os trabalhos de MARTÍNEZ-SANTIAGO et al. e de NETZER apresentam uma forma de sugerir candidatos conforme os símbolos ou pictogramas previamente selecionados pelo usuário. O primeiro faz uso de um preditor semântico em combinação com a Ontologia, enquanto que o segundo utiliza somente a Ontologia

como base para restringir os símbolos semanticamente relacionados. Possuir este critério permite uma melhoria na taxa de comunicação, uma vez que restringir o conjunto de pictogramas às opções com relação semântica reduz o tempo de navegação no SCAA. Além disso, possuir este critério facilita a utilização do SCAA, pois o usuário não terá mais a necessidade de navegar entre pastas e categorias para encontrar o pictograma desejado. No que se refere ao critério C4 (Implementação de Ferramenta) observa-se que todos os trabalhos apresentam uma interface que remete ao protótipo da proposta, entretanto, não foi encontrado evidência da existência de uma ferramenta para utilização. Ter acesso a ferramenta ou entender os detalhes de implementação permite a replicação da proposta. Observa-se pelo critério C5 (Utilização de *Word Bank*) que somente o trabalho de PAHISA-SOLé utiliza um *Word Bank* com as informações semânticas. Os outros trabalhos utilizam somente o conhecimento da Ontologia. Fazer uso de um *Word Bank* ou uma mesclagem com uma Ontologia pode permitir um processamento de informação eficiente, pois este tipo de solução se trata de aplicativos que podem ser utilizados em dispositivos móveis com recursos reduzidos.

Tabela 5 – Análise comparativa dos trabalhos relacionados

	Martinez-Santiago et al. (2016)	Netzer et al. (2006)	Pahisa-Solé (2012)
C1 - Léxico Personalizado	+	+	-
C2 - Gramática Visual	-	-	+
C3 - Sugestão de Candidatos	+	+	-
C4 - Implementação de Ferramenta	-	-	-
C5 - Utilização de Word Bank	-	-	+

Fonte: do Autor.

Diante desta análise, o trabalho proposto tem como objetivo atender à todos os critérios de avaliação de modo a melhorar em alguns aspectos específicos, detalhados a seguir. Para o primeiro critério, em nenhum trabalho têm-se a especificação de como o léxico personalizado é gerado a medida em que o conteúdo é editado pelo mediador. Neste sentido, espera-se apresentar uma melhora no que se refere a especificação de um gerador de léxico personalizado, que faz uso das alterações do léxico personalizado pelo mediador e alimenta o SCAA. Com relação ao segundo critério, a utilização de uma gramática visual pode trazer benefícios a uma criança em fase de alfabetização na educação especial. Utilizar deste recurso permite que o usuário associe cores às pictogramas e, conseqüentemente, à uma ordem de montagem da frase. Este guia visual está presente em apenas um dos trabalhos relacionados, entretanto, este guia não apresenta nenhuma forma de conduzir a sequência da frase, dado que considera que o usuário saiba o que está montando. Diante do terceiro critério, a sugestão de pictogramas candidatos atrelado à um guia de cores reforça a ideia de associação de conceitos aos pictogramas. Das duas propostas que oferecem esta sugestão, nenhuma leva em consideração esta associação. No que se refere a implementação da ferramenta, esta proposta pretende oferecer a especificação e o detalha-

mento de como um *parser* semântico para CAA pode ser criado. Além disso, este *parser* semântico pode ser baixado e testado, o que não ocorre nos trabalhos relacionados. Com relação ao último critério, trazer o benefício de um *Word Bank* pode reduzir o tempo e a quantidade de processamento necessários em análises sintática-semântica. Isto se torna importante quando as soluções são voltadas para dispositivos móveis, os quais podem variar em *hardware* e, conseqüentemente, em poder de processamento. Diante disto, no próximo capítulo apresenta-se a definição desta proposta, bem como a sua especificação arquitetural e funcional, seguido de uma avaliação de protótipo e comparativa com os trabalhos relacionados e os critérios elencados.

4 PROPOSTA

Neste capítulo apresenta-se o *parser* semântico para SCAA proposto neste trabalho. Primeiramente mostra-se uma visão geral onde são abordadas questões do cenário ideal e atual da proposta. Na sequência, faz-se um detalhamento dos componentes e suas funções. Em seguida, apresenta-se a visão funcional que detalha as atividades realizadas por cada componente, assim como a sequência em que estas ocorrem. Por fim, apresenta-se a avaliação prática do protótipo, seguido da avaliação conceitual e terminando com as considerações finais.

4.1 VISÃO GERAL

Para melhor compreender esta proposta e os requisitos que a guiaram, esta seção apresenta uma visão geral da interação do usuário em um cenário ideal. Em suma, a proposta visa especificar e implementar um *parser* semântico no contexto de CAA. Este *parser* semântico visa guiar, de forma visual e em tempo de execução, a criação de frases telegráficas em ordem direta e bem formadas. Para fins de exemplificação, esta proposta será implementada no aplicativo aBoard (LIMA et al., 2017), o qual é o SCAA gratuito de melhor avaliação (4,7 estrelas) na loja da Google¹. O aplicativo aBoard, assim como outros aplicativos do segmento, possui limitações referentes à criação de frases telegráficas bem formadas. Diante disto, o aBoard não oferece uma forma de guiar visualmente a construção de suas frases, permitindo que estas sejam construídas sem uma ordem correta. Por não restringir semanticamente a seleção dos pictogramas, o aplicativo permite a criação de frases sem significado. De modo a eliminar estas limitações, o *parser* semântico deve ser utilizado em conjunto como um sistema SCAA. Neste sentido, para cada pictograma escolhido pelo usuário, a interface deve se reorganizar de modo que facilite a seleção de futuros pictogramas. Para cada pictograma selecionado o sistema realiza a validação do estado atual da frase em nível sintático-semântico, garantindo uma expressão bem formada em linguagem telegráfica. Posteriormente, a frase é vocalizada por um recurso disponível no aplicativo. Vale ressaltar que uma vez que o aplicativo está em fase de reestruturação, para dar suporte a esta proposta, criou-se um sistema alternativo para a entrada de dados. Este sistema substitui o conjunto de pictogramas por uma lista de palavras que fazem referência aos pictogramas. Por se tratar de um sistema alternativo, este não representa a aplicação completa e pronta para utilização pelo usuário, apenas simula sua utilização.

Na Figura 28 mostra-se a interface na qual o usuário interage com o dispositivo. Esta interface é dividida em duas áreas: (A) área de expressão e (B) área de conteúdo. A área de

¹ Disponível em: <https://play.google.com/store/apps/details?id=assistive.aboard&hl=pt_BR>

Figura 28 – Interface do aBoard no cenário ideal

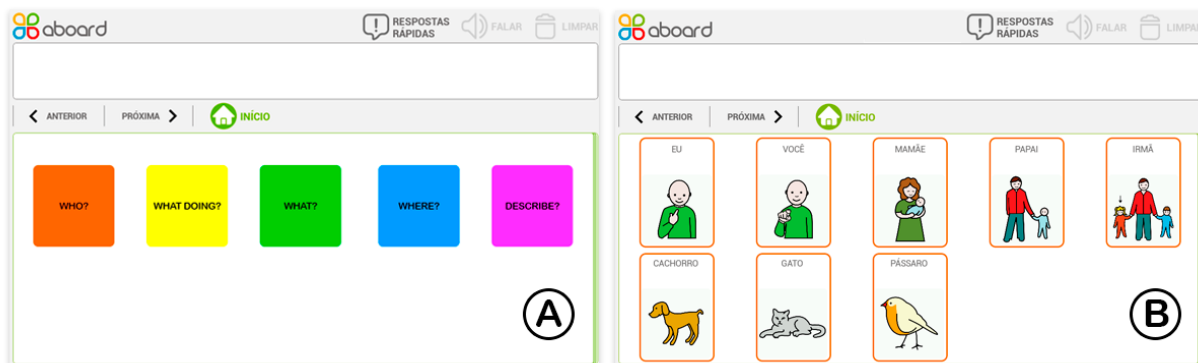


Fonte: do Autor.

expressão apresenta, de maneira visual, o estado atual da expressão e organiza os pictogramas em ordem direta de acordo com o CS. Na área de conteúdo tem-se as opções elegíveis pelo usuário inicialmente representada pelo padrão de cores do CS. Posteriormente, com a escolha de uma cor, a interface se modifica apresentando os pictogramas candidatos para seleção. Na Figura 29 ilustra-se esta dinâmica. Ao inicializar o sistema apresenta-se a tela (1), onde são oferecidas as opções de cores do CS a serem selecionadas. Após a seleção da cor, o sistema troca para a tela (2) que mostra os pictogramas candidatos. Ressalta-se que a tela (2) aparece em decorrência da seleção de uma das opções disponíveis em (1), por esse motivo, o conjunto de pictogramas mostrados ao usuário variam conforme a cor escolhida. No caso desta figura, foi escolhido a cor laranja (i.e., *Who?*) em (1), o qual foi direcionado para a tela (2) que apresenta os candidatos desta cor (i.e., quem pode ser o sujeito da frase).

Na Figura 30 apresenta-se um exemplo com o passo a passo para a construção da frase telegráfica “*I drink cold water*”. A primeira interação do usuário ocorre na seleção da opção laranja (*Who?*) que, em seguida, altera a área de conteúdo para exibir pictogramas relacionados à opção. Estes pictogramas levam em consideração sua categoria gramatical e função sintática (i.e., substantivos e pronomes referindo-se ao sujeito). Uma vez que o pictograma “*I*” é selecionado, o dispositivo altera a tela para mostrar as opções de cores. Assumindo que o usuário irá escolher a opção “*What Doing?*”, o sistema reorganiza a área de conteúdo a fim de preencher com pictogramas que definem verbos. Ressalta-se que estes pictogramas devem ter uma relação semântica com o pictograma previamente selecionado “*I*”, o que resulta em ações que somente uma “pessoa” poderia realizar (e.g., *eat*, *drink*, *sleep*). Seguindo o exemplo, após selecionar o pictograma que representa a

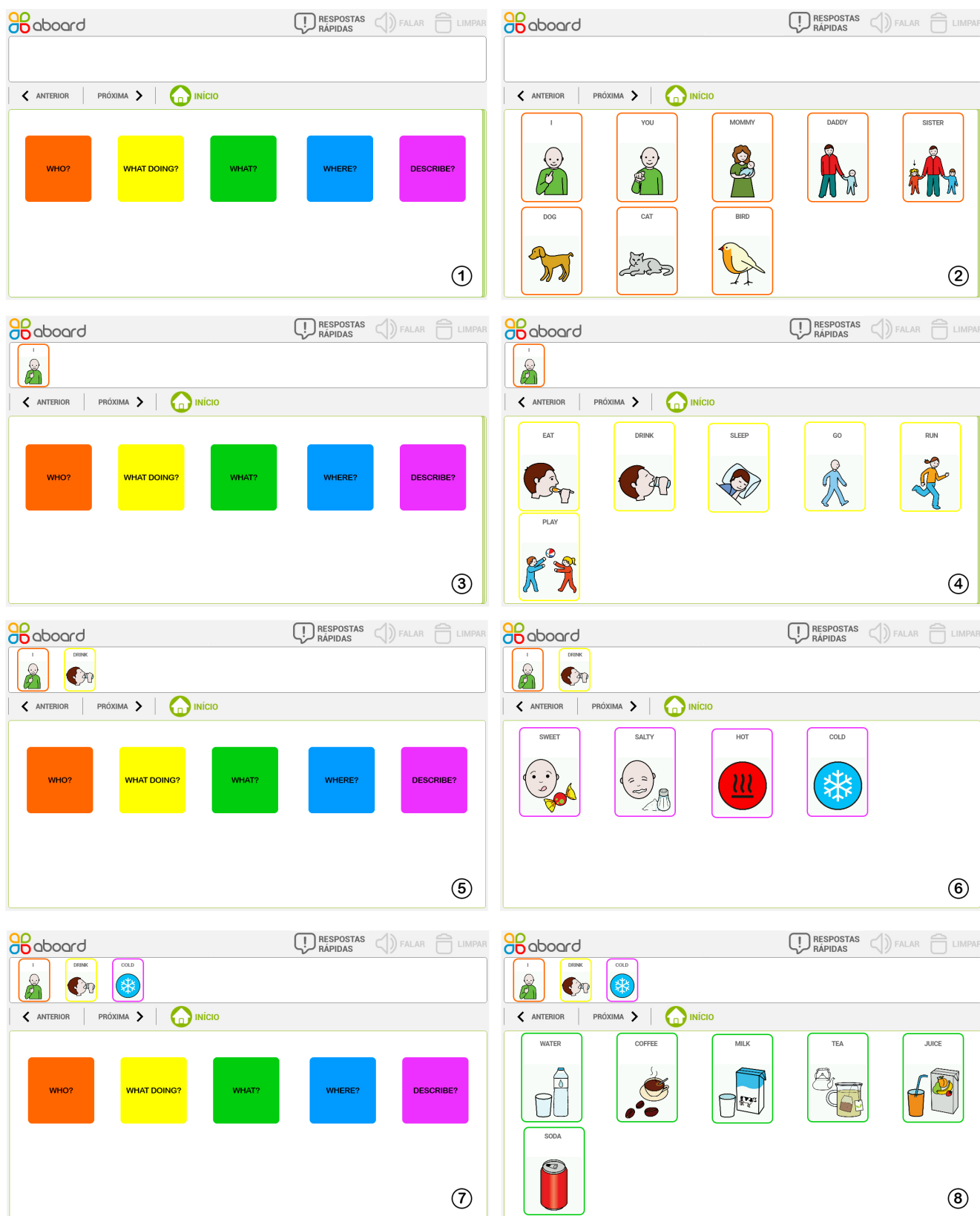
Figura 29 – Cenários de interação do usuário



Fonte: do Autor.

palavra “*drink*”, o dispositivo retorna para a tela das opções de cores. A seguir, seleciona-se a cor roxa, que designa um adjetivo (*Describe?*) e baseando-se na última escolha de verbo “*drink*”, o sistema identifica que deve mostrar ao usuário adjetivos que se relacionem semanticamente com algo bebível/líquido (e.g., *cold*, *hot*, *salty*, *sweet*). Após selecionar o pictograma “*cold*” o sistema altera novamente a tela para seguir ao próximo passo da construção da frase, a seleção da opção “*What?*”. Com base na relação semântica imposta do verbo *drink*, os pictogramas candidatos mostrados ao usuário referenciam líquidos. Com as restrições acontecendo em tempo de execução, garante-se que a frase construída pelo usuário seja bem formada. Ressalta-se que a opção “*Describe?*” é mais flexível, uma vez que pode se referir ao adjetivo do sujeito, objeto e lugar. Desta forma, a referência semântica para consulta de candidatos leva em consideração a última opção selecionada (e.g., se após selecionar a opção “*Who?*” o usuário selecionar a opção “*Describe?*”, este estará se referindo a um adjetivo do sujeito “*I*”).

Figura 30 – Visão geral do funcionamento do sistema para a construção da sentença “I drink cold water”



Fonte: do Autor.

4.2 ADAPTAÇÕES

Para que esta proposta atinja os objetivos propostos, é necessário adaptar a Teoria SLIM e sua implementação tecnológica, o JSLIM. Estas mudanças são apresentadas e explicadas a seguir.

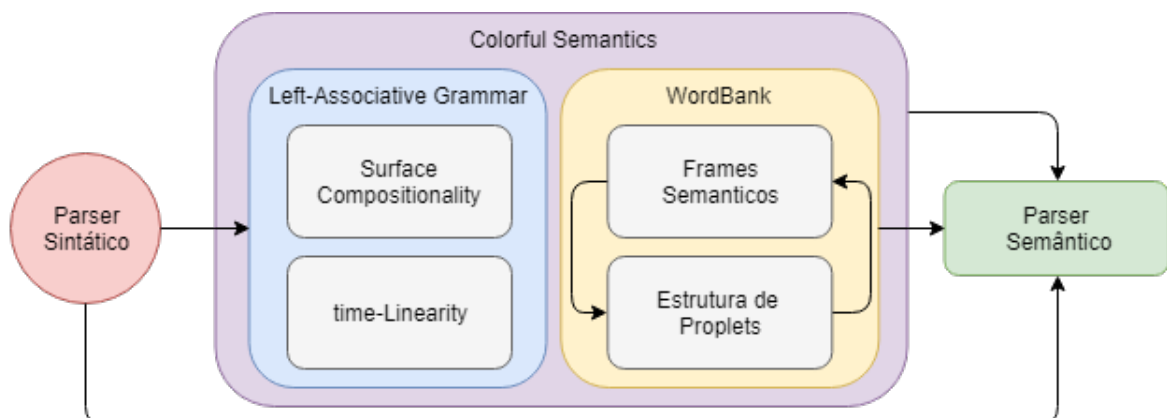
4.2.1 Adaptação Científica

Dado que a Teoria SLIM não tem o conceito de *parser* semântico para CAA, foi necessário realizar alterações de modo que esta contemple o novo conceito. Na Figura 31 apresenta-se a adaptação da teoria comparando-a a Figura 13 apresentada na Seção 2.4. Como pode ser observado, o modo “*hear*” e o modo “*speak*” foram adaptados, assim como a forma de realizar o *Internal Matching*.

No que se refere ao modo “*hear*”, definido originalmente pela teoria, este foi transformado em um *parser* sintático. Com essa transformação, o *parser* sintático se aproveita da ideia inicial de utilizar os padrões da teoria original (i.e., *Surface-compositionality* e *time-Linearity*) em combinação com a validação *Internal* e as regras de *Matching*, oferecido pelo CS (i.e., ordem direta).

O modo “*speak*” foi transformado em um *parser* semântico. Neste sentido, após a análise sintática, o *parser* semântico entra em ação. Para realizar esta interação, o *parser* semântico utiliza a ideia original do *WordBank* estruturado por *proplets*. Além disso, estes *proplets* foram enriquecidos com informações semânticas provenientes de *frames* semânticos e são também guiados pelo conceito do CS, indicando qual é a relação semântica do constituinte seguinte da frase.

Figura 31 – Adaptação da Teoria SLIM



Fonte: do Autor.

4.2.2 Adaptação Tecnológica

A tecnologia utilizada para atingir os objetivos da proposta foi o JSLIM 2.0. Entretanto, assim como a teoria, esta ferramenta não implementa um *parser* semântico, tornando necessário algumas adaptações para atingir os objetivos da proposta. Destaca-se que foi necessário analisar toda a ferramenta, realizar testes e *debugs*, pois não há documentação referente ao código fonte.

A estrutura principal do JSLIM possui como arquivo de configuração principal a extensão *.pro* (cf. Figura 32). Este arquivo armazena informações referentes a localização de todos os outros arquivos de configuração:

- **Regras Gramaticais:** arquivo com a extensão *.rul* que possui as regras gramaticais para validação morfológica e sintática;
- **Regras de Flexão:** arquivo com a extensão *.tbl* que possui regras de flexão do léxico ou regras dinâmicas que identificam se o elemento do léxico é um verbo, substantivo ou adjetivo;
- **Léxico:** arquivo com a extensão *.lex/.all* que armazena o léxico do sistema em formato de *template*, fazendo alusão aos *proplets*;
- **Atributos:** arquivo com a extensão *.att* que contém os atributos e o tipo de dado disponíveis dentro da estrutura de *template* do arquivo de léxico.

Figura 32 – Exemplo de arquivo de projeto JSLIM

```
Project file
enc: UTF-8
att: english1.att
ops: english1.ops
tbl: ./filter.tbl

lamorph
rul: ../morph/combi.rul
var: ../morph/english1.var
lex: ../allo/allo.all
sem: ../morph/sem.lex

lahear
rul: ../syn/syntax.rul
var: ../syn/syntax.var
```

Fonte: do Autor.

As adaptações realizadas nos arquivos afetaram somente os arquivos de variáveis e léxico. O primeiro, conforme mostrado na Figura 33, refere-se a inclusão do atributo

especial “*ontology*” em formato lista, com isto, o JSLIM identifica que este é um atributo válido para ser interpretado no arquivo de léxico, dentro da estrutura de *template*. Na Figura 34, apresenta-se a estrutura de *template* dentro do arquivo de léxico, contendo o atributo especial “*ontology*”. Como pode ser observado, o *template* é definido pela palavra reservada “*!template*”, esta atribui todos os pares de valores de atributo entre colchetes para todas as entradas de léxico. A vantagem desta notação é de que os léxicos de entrada que compartilham da mesma estrutura são declarados somente uma vez. Desta forma, o atributo especial “*ontology*” se torna essencial na adaptação, pois armazena as informações semânticas (i.e., classe semântica que referencia a classe na ontologia) de cada palavra do léxico, ou seja, a palavra “*eat*” pode possuir em seu atributo especial “*ontology*” a classe semântica “*EatingAction*”.

Figura 33 – Exemplo de arquivo de atributos .att

```

string  sur
        core
        noun
        verb
        adj
list    cat
        sem
        ontology
table   allo
        flx
        syn
list    mdr
        mdd
        fnc
        arg

```

Fonte: do Autor.

Conforme visto na Seção 2.4.2, o Word Bank armazena as relações sintáticas e semânticas entre todas as instâncias. Na abordagem atual, a adaptação realizada abstrai este detalhamento de modo que as restrições sejam definidas pelos frames semânticos existentes na Ontologia. A diferença das abordagens pode ser vista na Figura 35, sendo (A) a abordagem original e (B) a abordagem atual. Em (A) tem-se a replicação das informações semânticas para cada palavra, enquanto que em (B), armazena-se os nomes dos *frames* da ontologia, indicando onde as relações semânticas são encontradas (e.g., com base na Figura 35, em (B), a palavra “*Daddy*” tem relação com os *frames* “*EatingFrame*” e “*DrinkingFrame*”). Para isso, criam-se duas novas tabelas: tabela de classes semânticas e tabela de frames. A tabela de classes semânticas tem um relacionamento de (n:n) com o propleto,

Figura 34 – Exemplo de arquivo de léxico .all

```

# verbs
!template[cat: (n' v), syn: T_1]
![sur, core, ontology]
drink drink (DrinkingAction)
eat eat (EatingAction)

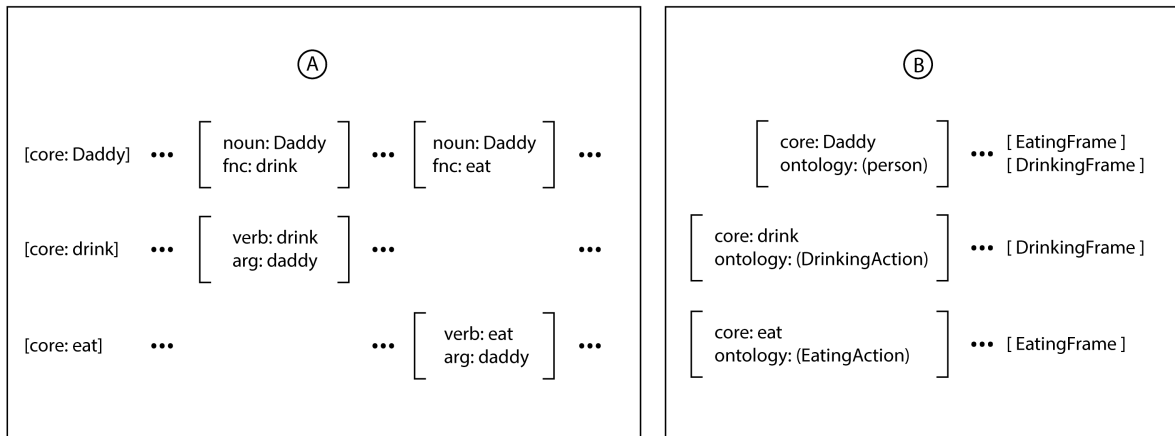
# nouns + pronouns
!template[cat: (sn), syn: T_2]
![sur, core, ontology]
daddy daddy (Person)
cat cat (Animal)

# adjectives
!template[cat: (adj), syn: T_3]
![sur, core, ontology]
orange orange (ColorAttribute)
big big (SizeAttribute)

```

Fonte: do Autor.

o que significa que um *proplet* pode ser definido por uma ou mais classes semânticas. Na tabela de *frames*, tem-se uma relação de (n:n) com a tabela de *tokens* (cf. Seção 2.4.2), o que significa que um *token*, definido por um *proplet*, possui uma ou mais relações com algum *frame* existente na Ontologia. Diante disto, a forma de navegação no *Word Bank* que define “*what to say*” (i.e., LA-Think) e a forma de estruturar a frase definido por “*how to say*” (i.e., LA-Speak), explicados na Seção 2.4.2, se tornam diferentes. Para a forma de navegação *Word Bank*, utiliza-se a combinação das classes semânticas do *proplet*, a relação de *frames* semânticos do *proplet* e a opção do CS escolhida (cf. Seção 4.4). A forma com que a estrutura da frase deve ser montada é definida pela gramática visual do CS, isto é, a ordem da frase (“*how to say*”) segue a ordem direta conforme a sequência de cores do CS. Na Seção 4.4 encontram-se detalhes deste funcionamento.

Figura 35 – Diferença da estrutura de *proplet* original e proposto

Fonte: do Autor.

Com a mudança na estrutura de *proplets* armazenados no DBS, o arquivo que será lido

pelo JSLIM para inserir as informações de léxico e suas restrições sintáticas e semânticas é um arquivo denominado “dbs.sql” (cf. Figura 36). Este arquivo é produto do componente Gerador do Léxico Personalizado, sendo este descrito nas Seções 4.3 e 4.4.

Figura 36 – Arquivo de inserção do léxico e restrições sintáticas e semânticas no DBS

```
INSERT INTO TYPE (id, baseform, pos) VALUES (1, 'drink', 'verb');

INSERT INTO PROPLET (id, core) VALUES (1, 'drink');

INSERT INTO SEMANTIC (id, name) VALUES (1, 'DrinkingAction');

INSERT INTO SEMANTIC_RELATION (semantic_id, proplet_id) VALUES (1, 1);

INSERT INTO TOKEN (type_id, proplet_id) VALUES (1, 1);

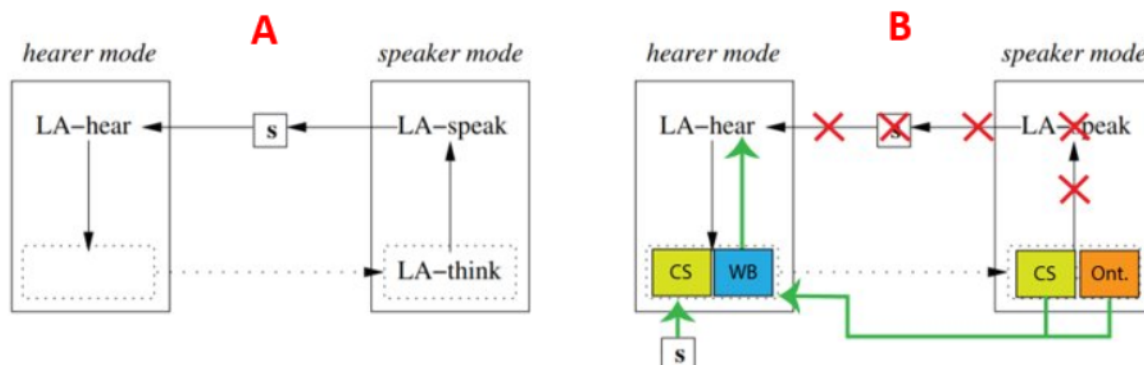
INSERT INTO FRAME (id, name) VALUES (1, 'DrinkingFrame');

INSERT INTO FRAME_RELATION (type_id, frame_id) VALUES (1, 1);
```

Fonte: do Autor.

Diante destas alterações, na Figura 37 apresenta-se um comparativo da versão do DBS original (A) com a nova versão (B). Na versão original, os componentes LA-Hear e La-Speak se comunicavam, de modo que o primeiro interpretava as frases criadas pelo segundo. Estas frases são criadas com o auxílio do LA-Think, que tem o objetivo de consultar qual conteúdo da frase que pode ser produzido. Com a nova adaptação, os componentes LA-Speak e LA-Think foram removidos, dando lugar a combinação do CS com a Ontologia. Esta combinação serve para guiar a navegação e a obtenção dos pictogramas candidatos no *WordBank* e na Ontologia. Além disso, o La-Hear foi transformado em um *parser* sintático-semântico, que faz uso do *WordBank* em combinação com as regras do CS. Ambas as adaptações se comunicam de forma bi-direcional. Na Seção 4.4 apresenta-se uma melhor explicação de como esta adaptação funciona na prática.

Figura 37 – Comparação entre o DBS original e o Adaptado



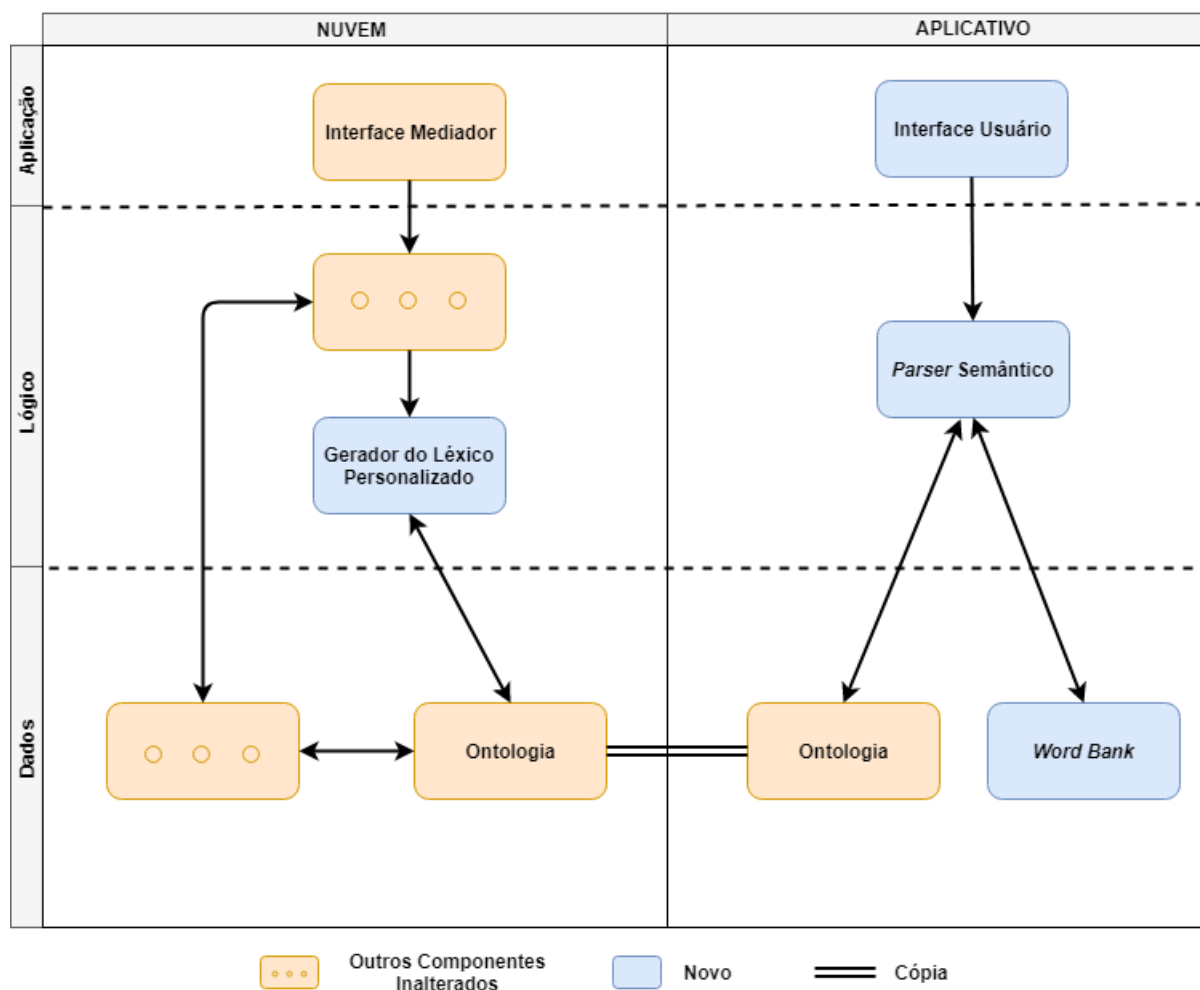
Fonte: do Autor.

4.3 ARQUITETURA

Visando integrar o trabalho proposto ao sistema aBoard, na Figura 38 apresenta-se uma arquitetura geral de *software* dividida em dois módulos: gerador do léxico personalizado e *parser* semântico. Esta arquitetura possui módulos que são utilizados em ambientes diferentes, o primeiro (i.e., gerador de léxico personalizado) deve ficar na nuvem e o segundo (i.e., *parser* semântico) implementado no aplicativo. Porém, o sistema aBoard ainda não possui uma estrutura compatível com estes novos módulos. Sendo assim, os módulos foram implementados como um protótipo em um projeto único.

Com relação aos componentes marcados como inalterados, estes não fazem parte desta proposta. Entretanto, são responsáveis por todo o funcionamento do sistema em nuvem, exercendo tarefas como gerenciar a lógica da interface e o conteúdo personalizado do usuário. No que se refere ao sistema em nuvem, o componente inalterado (Interface Mediador) da camada de aplicação, tem o objetivo de oferecer uma interface para o mediador personalizar o léxico que estará disponível no aplicativo. É a partir desta interface que todas as ações (e.g., editar legenda do pictograma, excluir pictograma) do sistema em nuvem são realizadas. Os componentes inalterados da camada lógica e de dados fazem alusão a mais de um componente, isto é, eles representam diversos componentes que tem o objetivo de personalizar o aplicativo aBoard. Estas personalizações referem-se à mudanças de interface (e.g., quantidade de pictogramas por tela, tamanho das imagens, contraste) e de conteúdo (e.g., legenda de pictograma, quais pictogramas fazem parte do léxico do usuário). Na camada de dados, os componentes inalterados armazenam as informações personalizadas que o mediador configurou. Vale ressaltar que estes componentes inalterados não serão detalhados, uma vez que não fazem parte desta proposta e não prejudicam o entendimento da mesma. Quanto ao módulo Gerador de Léxico Personalizado, este é acionado ao final do procedimento de personalização do mediador. A ação de salvar as alterações feitas pelo mediador na interface gera uma requisição ao módulo Gerador de

Figura 38 – Arquitetura Geral da Proposta



Fonte: do Autor.

Léxico Personalizado. Este, por sua vez, se alimenta das informações modificadas para exercer sua função de criar arquivos contendo o léxico personalizado e as restrições semânticas (i.e., o verbo “*drink*” possui restrições semânticas que impedem que um alimento sólido seja seu complemento, assim como um objeto realize a ação determinada por este verbo).

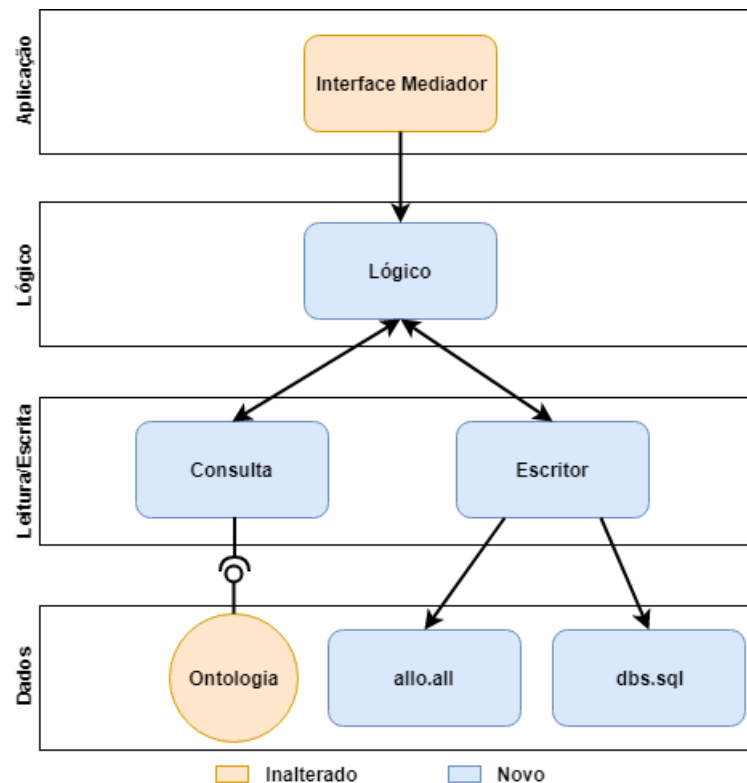
Com relação à arquitetura do aplicativo, o único componente inalterado diz respeito à Ontologia. Esta possui informações do léxico personalizado do usuário, sendo ela uma cópia da versão existente na nuvem. Os demais módulos que fazem parte da proposta serão detalhados a seguir.

4.3.1 Arquitetura Gerador do Léxico Personalizado

Uma das características da solução aBoard é dispor de uma ferramenta de criação de vocabulário em nuvem, que permite modelar um léxico personalizado. Este léxico é ex-

presso em uma Ontologia². Neste contexto, o módulo Gerador do Léxico Personalizado tem como objetivo obter e estruturar as informações referentes ao léxico, à gramática e à semântica para serem consumidos pelo *parser* semântico. Este módulo é composto por três submódulos: consultor, escritor e lógico. Vale ressaltar que o gerador não está acoplado ao *parser* semântico e será implementado na nuvem junto com a ferramenta de criação do vocabulário personalizado. Desta forma realiza-se um pré-processamento dos dados em um sistema mais robusto (i.e., servidor) para utilizar em um dispositivo móvel. Na Figura 39 mostra-se a arquitetura deste módulo, constituído por 4 camadas: Aplicação, Lógica, Leitura/Escrita e Dados. Ressalta-se que a interface do mediador assim como a ontologia não fazem parte da proposta.

Figura 39 – Arquitetura do Gerador do Léxico Personalizado



Fonte: do Autor.

Partindo da camada lógica, o submódulo lógico define regras de funcionamento do gerador, seu objetivo é intermediar o fluxo de dados dos submódulos consultor e escritor. Para isto, este submódulo realiza requisições ao submódulo consultor de modo a adquirir informações do léxico e de relações semânticas existentes na Ontologia. O resultado destas consultas servem como *input* de dados para o submódulo escritor, que salva esses dados em arquivos que posteriormente serão reconhecidos e utilizados pelo *parser* semântico (cf. Seção 4.4).

² Esta modelagem está em processo de desenvolvimento por outro trabalho acadêmico. O léxico é resultado da pesquisa de (FRANCO et al., 2017b)

Para a camada de leitura/escrita tem-se os submódulos consultor e escritor. O submódulo consultor oferece ao submódulo lógico uma interface para conexão e consulta à Ontologia. Estas consultas oferecem informações referentes as instâncias, restrições sintáticas e semânticas do vocabulário personalizado do usuário (i.e., considerando a palavra “eat, esta possui restrições sintáticas que dizem respeito ao seu complemento ou o seu sujeito, assim como restrições semânticas que limitam a alimentos comestíveis). No que se refere ao submódulo escritor, este realiza ações de criação de dois arquivos, os quais estão representados na camada de dados (i.e., allo.all e dbs.sql). A estrutura do arquivo allo.all e dbs.sql seguem os formatos apresentados na Seção 4.2.2.

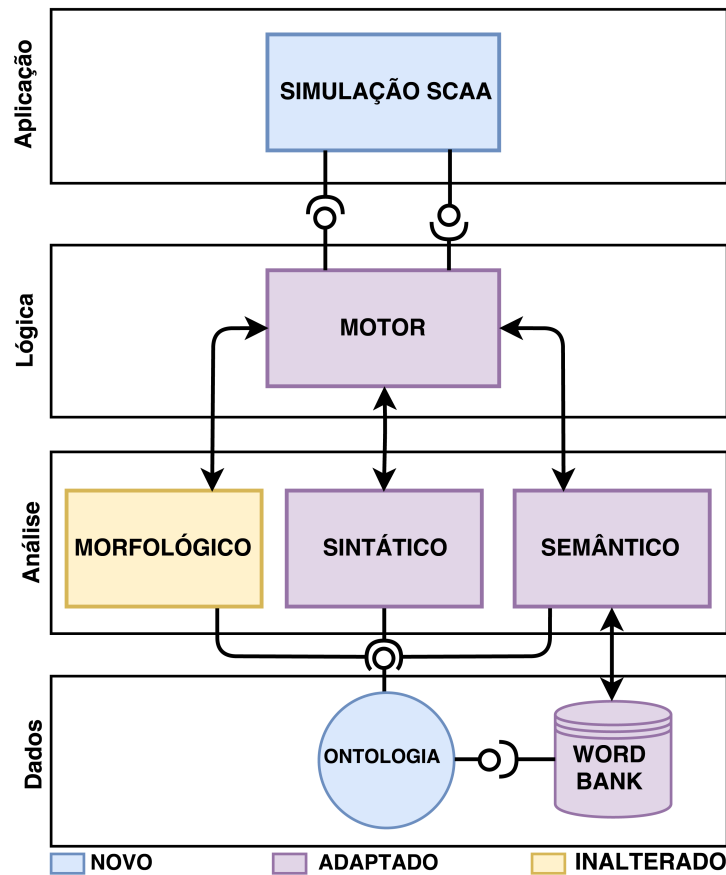
4.3.2 Arquitetura Parser Semântico

Na Figura 40 apresenta-se a arquitetura do *parser* semântico, o qual é dividido em 4 camadas: Aplicação, Lógica, Análise e Dados. A camada de aplicação é a parte visível ao usuário, que tem o objetivo de fornecer uma interface para *input* dos dados, bem como para visualização e interação com os resultados provenientes das camadas inferiores. A camada lógica contém o componente *Motor*, o qual é responsável por identificar, direcionar e ativar, quando necessário, cada módulo da camada de análise. A camada de análise é dividida em três módulos: morfológico, sintático e semântico. Estes módulos tem o objetivo de realizar a análise e a validação do pictograma escolhido pelo usuário, assim como obter os pictogramas candidatos. Por fim, a camada de dados é responsável por fornecer as informações linguísticas e de domínio do sistema, isto é, contém o léxico e as informações morfológicas, sintáticas e semânticas necessárias para a camada de Análise.

Para a camada de aplicação tem-se o componente Simulador SCAA que visa reproduzir as ações do cenário ideal. Este componente tem o objetivo de oferecer uma interface em modo texto de interação ao usuário. Com ela, o usuário poderá navegar entre as opções de cores do CS assim como as sugestões de constituintes oferecidas pelo *parser* semântico de forma dinâmica. Desta forma, este componente é visto como a parte de entrada e saída de dados do usuário.

O componente da camada lógica é chamado de *Motor*, este intermedia os estados de análise, recebendo os dados inseridos pelo usuário e determinando a sequência das etapas de análise de forma automatizada. Neste sentido, ao finalizar cada análise, o *Motor* alterna para a próxima etapa de validação até obter os pictogramas candidatos e retorná-los ao usuário ou as opções de cores do CS.

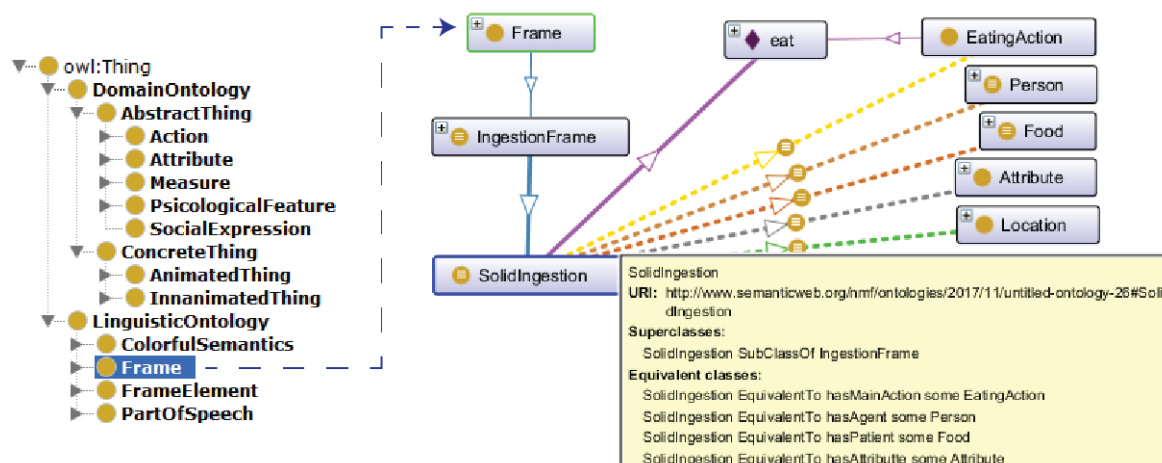
A terceira camada é composta por três componentes de análise: morfológico, sintático e semântico. o componente morfológico tem o objetivo de realizar a análise morfológica do *input* do usuário no momento em que um constituinte candidato for escolhido, fazendo sua identificação dentro da estrutura de *template* definido anteriormente na Seção 4.2.2 e construído pelo Gerador de Léxico Personalizado (cf. Seção 4.4). O componente sintático realiza a análise sintática, que refere-se a segunda etapa da análise. Esta deve aplicar as

Figura 40 – Arquitetura do *parser* semântico

Fonte: do Autor.

regras gramaticais, utilizando o formalismo LAG (cf. Seção 2.4.1), existentes no arquivo “*syntax.rul*” (cf. 4.2.2) para o candidato escolhido pelo usuário. O terceiro componente tem o objetivo de validar semanticamente a opção do usuário, além disso, deve obter informações sintáticas e semânticas desta escolha, utilizando o atributo especial “*ontology*” criado na adaptação tecnológica do JSLIM (cf. 4.2.2).

A camada de dados é constituída pelos componentes Ontologia e *Word Bank*. A Ontologia detém os conhecimentos de domínio e os conhecimentos linguísticos necessários para alimentar o *Word Bank* assim como prover o léxico do sistema. A estrutura da Ontologia pode ser vista na Figura 41, a qual utiliza de *frames* semânticos para definir as restrições semânticas de cada elemento do léxico do usuário. No componente *Word Bank* tem-se o espelhamento de algumas informações da Ontologia (i.e., instâncias, classes semânticas e o nome dos *frames* semânticos) em formato de *proplets* que facilitam a análise semântica da proposta (e.g., considerando a palavra “*eat*” armazenada no *Word Bank*, esta pode ter a relação com a classe semântica “*DrinkingAction*” e com o *frame* semântico “*DrinkingFrame*”).

Figura 41 – Exemplo de *frame* semântico na Ontologia

Fonte: do Autor.

4.4 VISÃO FUNCIONAL

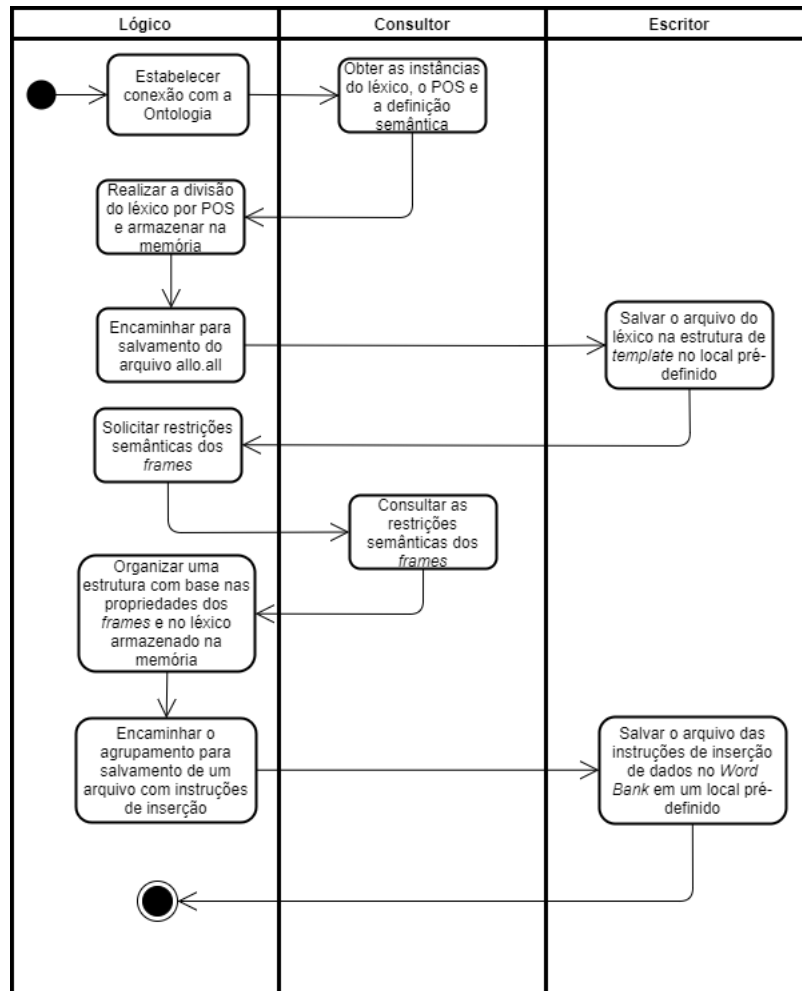
Nesta seção apresenta-se a visão funcional da proposta, onde são detalhadas as etapas de geração do arquivo do léxico e do arquivo para inserção de dados no *Word Bank*, no componente Gerador de Léxico Personalizado. Para o componente *parser* semântico, são detalhadas as atividades realizadas para a validação da frase e obtenção de cada pictograma candidato.

4.4.1 Visão Funcional do Gerador do Léxico Personalizado

Na Figura 42 apresenta-se o diagrama de atividades do componente Gerador do Léxico Personalizado no processo de criação dos arquivos necessários para os dados de entrada do *parser* semântico, bem como o arquivo de inserção de dados no *Word Bank*. No cenário ideal, o fluxo se inicia no momento em que o usuário mediador finaliza a especificação do vocabulário na ferramenta de criação disponível na nuvem. Para isto, o Gerador do Léxico Personalizado é ativado, pré-processando o léxico e as definições e ligações semânticas (e.g., dado a palavra “Daddy”, este tem como definição semântica a classe “*Person*”, que por sua vez pode possuir ligações semânticas com os *frames* “*PersonDrinkingFrame*” e “*PersonEatingFrame*”). Como explicado anteriormente, esta proposta trata-se de um protótipo, portanto, não está, de fato, na nuvem. Neste sentido, a etapa de pré-processamento é realizada após a confirmação do usuário via comando por texto.

No cenário atual, o Gerador se conecta à Ontologia para obter todas as instâncias (i.e., palavras) existentes, sua classe gramatical e sua classe semântica. Ressalta-se que estas instâncias se referem ao vocabulário personalizado do usuário. Em seguida, o submódulo lógico realiza o agrupamento destes resultados usando como parâmetro a função sintática de cada instância. Estes grupos são organizados com base nas funções sintáticas

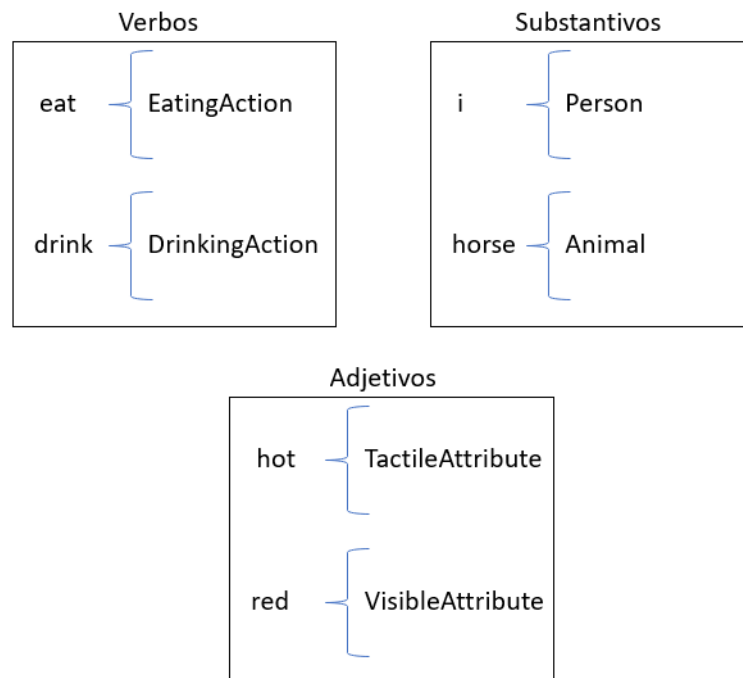
Figura 42 – Diagrama de Atividades do Gerador



Fonte: do Autor.

(e.g., sujeito, verbo, objeto) do CS, diante disto, criou-se os grupos de substantivos (i.e., pronomes e substantivos), verbos e adjetivos. Vale destacar que as instâncias com mais de uma classe gramatical (e.g., *orange*, *fish*) estarão presentes em mais de um grupo. Após o agrupamento, o gerador salva esses grupos em memória e autoriza o submódulo escritor a salvar um arquivo com o léxico. O submódulo escritor, em seguida, faz uso das instâncias agrupadas para criar três arquivos nomeados pela classe gramatical (i.e., noun.lex, verb.lex, adj.lex) (cf. Figura 43), para, na sequência, mesclar esses arquivos e gerar um arquivo único denominado “*allo.all*”, o qual faz referência ao arquivo interpretável pelo JSLIM (cf. Seção 4.2.2). Dado que o foco da proposta é criar frases bem formadas, em linguagem telegráfica e na ordem direta, o pré-processamento dos alomorfos (i.e., variação de um morfema sem mudança no seu significado) do JSLIM não é realizado. Entretanto, este procedimento será utilizado em implementações futuras, a fim de contribuir na criação de um sistema COMPANSION.

Figura 43 – Exemplo de estrutura de verbos, substantivos e adjetivos



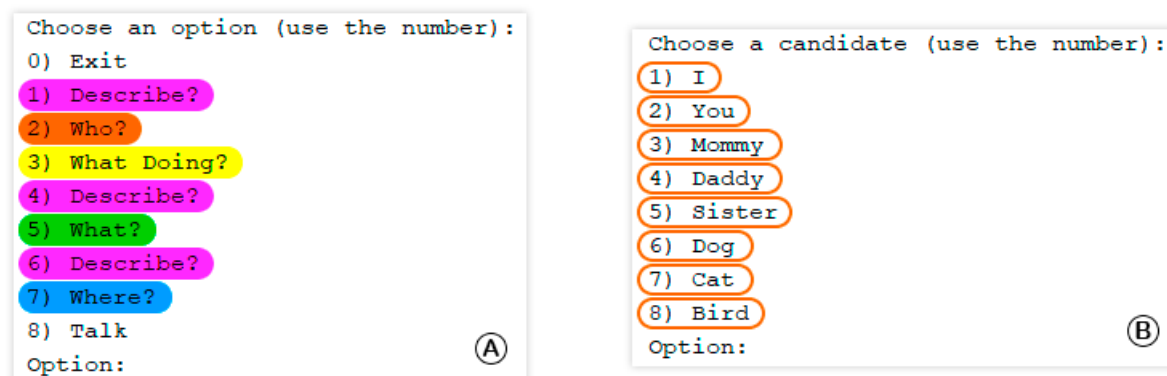
Fonte: do Autor.

4.4.2 Visão Funcional do *Parser* Semântico

A inicialização e configuração do *Parser* Semântico (PS) é feita em duas etapas: leitura/interpretação dos arquivos do projeto e inserção do conhecimento semântico no *Word Bank*. Assim, a primeira etapa de inicialização diz respeito a leitura do arquivo do projeto interpretando o arquivo do léxico (i.e., *allo.all*), dos atributos gramaticais (i.e., *assistive.var*), restrições sintáticas (i.e., *syntax.att*) e as regras gramaticais (i.e., *syntax.rul*). As estruturas destes arquivos foram detalhados na Seção 4.2.2. Após a leitura dos arquivos do projeto, realiza-se a segunda etapa de configuração, a qual faz a inserção do conhecimento semântico no *Word Bank*. Este conhecimento está no arquivo criado pelo componente gerador (cf. Figura 36). Ao fim, o usuário pode interagir com a interface do protótipo SCAA fornecendo os dados de *input* ao *parser* semântico. Na Figura 44 apresenta-se a equivalência das telas do cenário ideal (cf. Figura 29) em relação ao cenário atual, representado de forma textual. Na tela (A) mostra-se uma lista de opções numeradas e ordenadas seguindo o padrão CS de cinco níveis enquanto que, na tela (B) mostra-se a lista numerada dos pictogramas candidatos da cor escolhida.

Na Figura 45 apresenta-se um diagrama de atividades de modo geral, isto é, como acontece de forma resumida a interação do usuário com o PS. Estas atividades se dividem em 3 etapas de interação que posteriormente serão detalhadas, em um diagrama de atividades para cada etapa. Como ponto de partida (A), o usuário tem em sua interface as cores do CS para seleção. Ao escolher uma destas cores, o *parser* semântico deve encontrar

Figura 44 – Interface do simulador SCAA

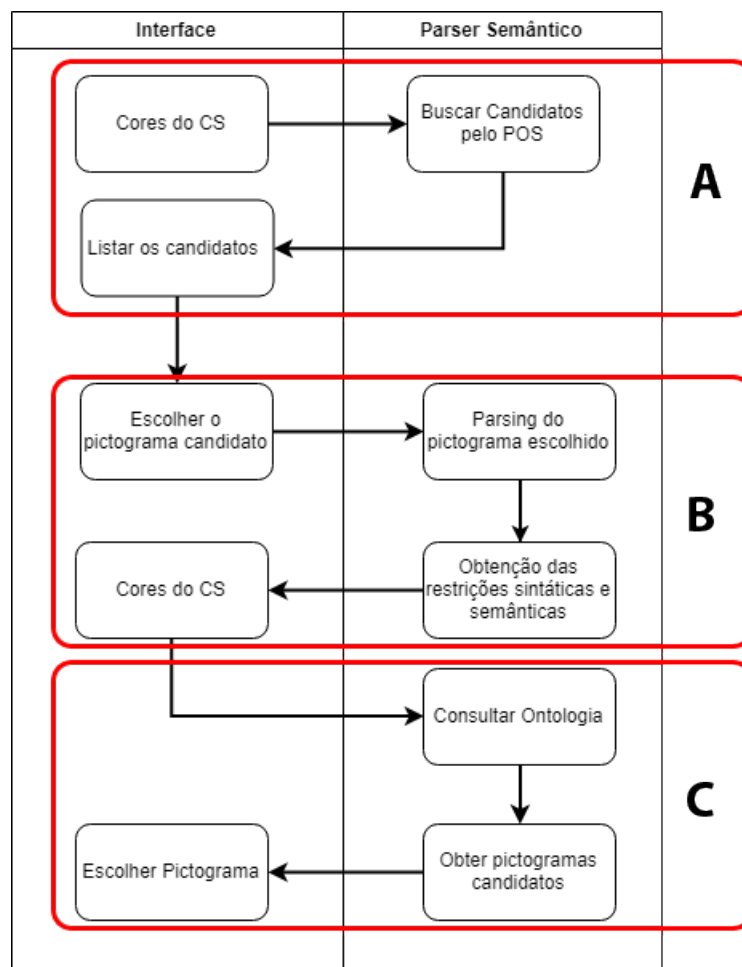


Fonte: do Autor.

os pictogramas candidatos com base na classe gramatical que esta cor representa (e.g., a cor laranja representa a opção “*who?*” que tem como classe gramatical um substantivo ou pronome). Assim, ao retornar os candidatos, faz-se uma listagem dos candidatos que se enquadram nesta classe gramatical. Em seguida, na etapa (B), o usuário realiza a ação de escolher o pictograma desejado, com isto, gera-se uma ação diferente da primeira etapa. O PS realiza a análise morfológica, sintática e semântica, de modo que valide este *input*, obtendo e armazenando informações das restrições sintáticas e semânticas devido a escolha do pictograma (e.g., escolhendo o pronome “*I*”, este carrega informações que dizem com quais *frames* semânticos este pictograma pode se relacionar). Para a etapa (C), considera-se que exista ao menos um pictograma selecionado, desta forma, ao escolher a próxima cor do CS. Sendo assim, PS considera as restrições sintáticas e semânticas do pictograma escolhido anteriormente e a cor escolhida, para utilizá-los como parâmetro de consulta e listagem dos novos candidatos. Assim, realiza-se uma consulta na Ontologia buscando pelas classes semânticas dos próximos dos candidatos que contemplem a classe gramatical da cor escolhida. Em seguida, utiliza-se os resultados como parâmetro de consulta ao *WordBank* de modo que encontre os candidatos. Ao final, retorna-se a lista de candidatos ao usuário que ao escolher uma opção, inicia um novo ciclo a partir da etapa B.

A seguir, será apresentado os diagramas de atividades de cada uma das três etapas. Para melhor compreender estas atividades, considera-se a construção da frase exemplo: “*I eat cake*”. Na Figura 46 apresenta-se o diagrama de atividades da etapa A da Figura 45. Esta etapa só é realizada na primeira ação do usuário, quando não há pictograma na área de construção de expressão. Considerando o exemplo, o usuário realiza a escolha da opção “*Who?*” - Laranja. Por ser a primeira interação, o componente *Motor* direciona o fluxo para a análise semântica, que realiza a solicitação dos pictogramas/palavras candidatos ao *Word Bank*. A obtenção dos candidatos no *WordBank* é simples pois, devido as

Figura 45 – Diagrama de Atividades Macro

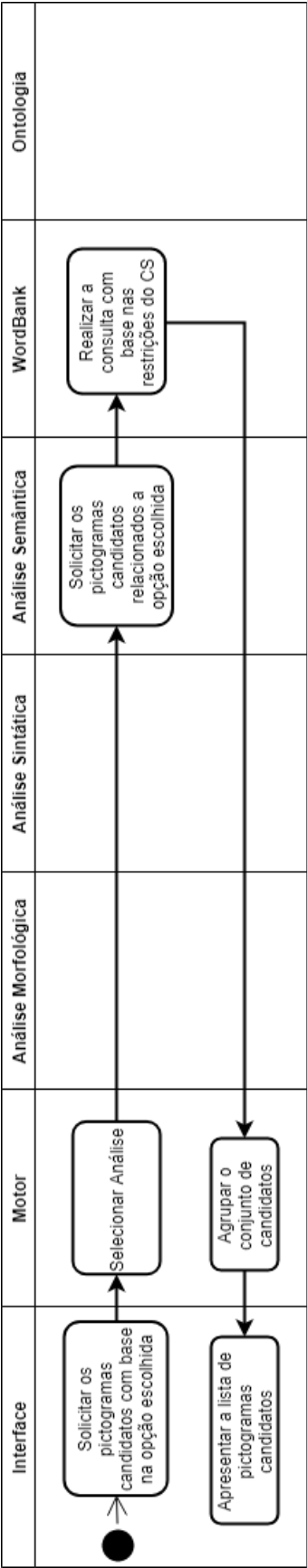


Fonte: do Autor.

facilidades oferecidas pelo padrão CS, é possível identificar que a opção "*Who?*" referencia um sujeito e este só pode ser definido por um substantivo ou pronome. Assim, realiza-se uma consulta ao *Word Bank* com o intuito de encontrar as palavras que atendam as restrições de sujeito (i.e., substantivo e pronomes). Os resultados desta primeira consulta são chamados de pictogramas candidatos. A partir desta lista, o *Motor* realiza o agrupamento dos candidatos e passa para a Interface SCAA, o qual desempenha a função de apresentar, em uma lista, cada um destes candidatos. Na Figura 47, apresenta-se um exemplo de como funciona a primeira etapa de consulta e listagem de candidatos. Ao escolher "*Who?*", busca-se por substantivos/pronomes no *WordBank*, que ao encontrar, retorna uma lista para o componente *Motor*, apresentando à Interface estes resultados (e.g., *I, Daddy, Mommy*).

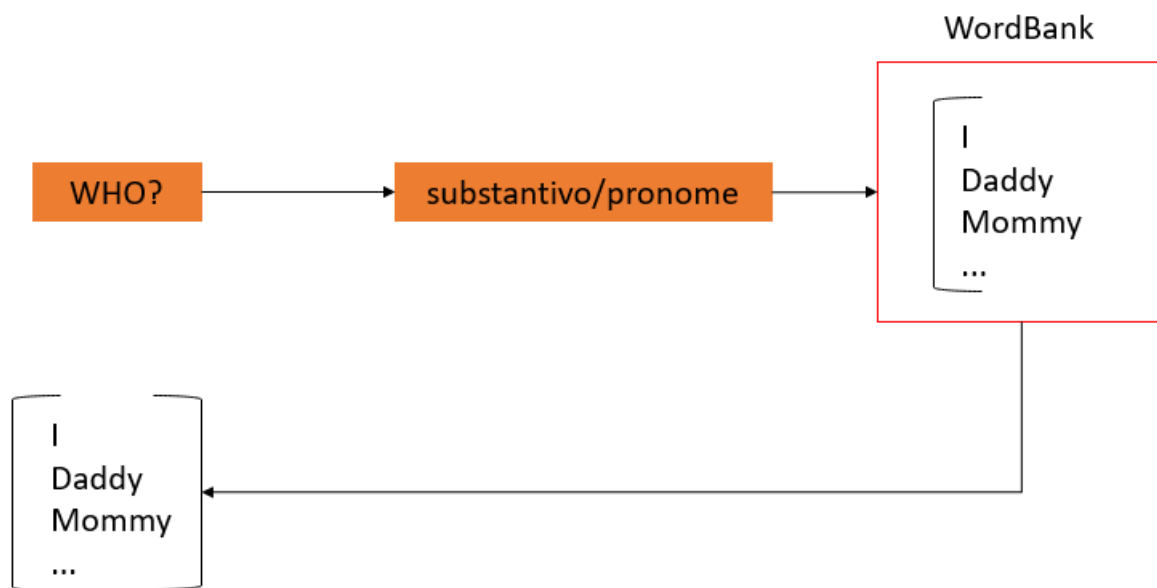
Na Figura 48, apresenta-se a etapa (B), onde o usuário realiza a ação de escolher o pictograma desejado. Ao receber o pictograma desejado pelo usuário, o sistema aciona o componente *Motor* que realiza a sequência de análises. Na análise morfológica, identifica-se a palavra, encontrando o atributo "*core*" no *template* que o define. Uma vez identificada a

Figura 46 – Diagrama de Atividades da Etapa A



Fonte: do Autor.

Figura 47 – Exemplo de busca por candidatos da Opção “Who?”

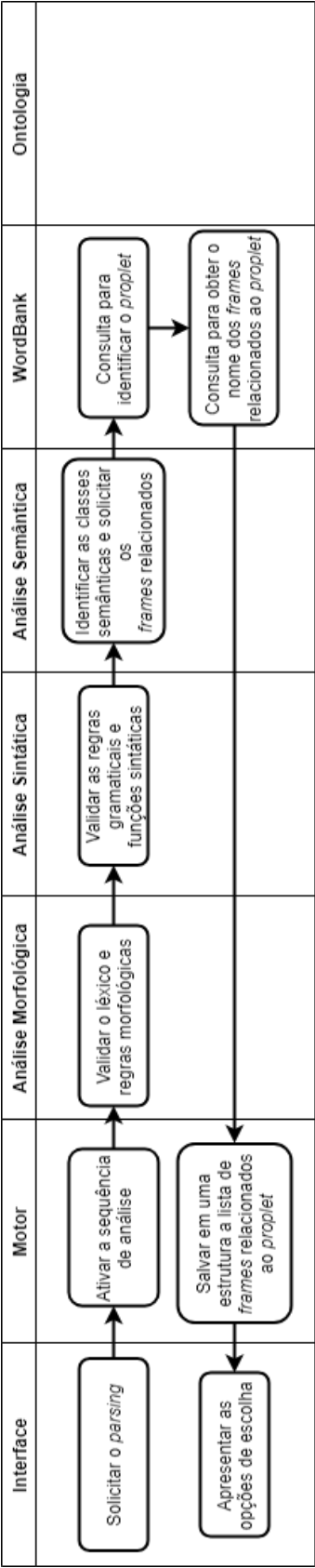


Fonte: do Autor.

estrutura, o *Motor* passa para a análise sintática. Nesta estrutura de *template* (e.g., Figura 49) tem-se os atributos “cat” e “syn” (e.g., quando se tratar de um verbo, o atributo “syn” recebe o valor “T_1”, que acrescenta dinamicamente atributos de verbo) que possuem informações sintáticas necessárias para esta análise. Estes atributos são explicados na Seção 2.4.2. Identificado os valores de “cat” e a estrutura referenciada pelo atributo “syn”, a análise sintática realiza a validação gramatical conforme as regras definidas no arquivo “syntax.rul” (e.g., considerando uma frase “*I drink*” e a definição de LAG na Seção 2.4.1, a frase inicial da análise é expressa pela palavra “*I*”. Esta realiza a verificação da próxima palavra “*drink*” de modo a identificar se esta frase está sintaticamente correta), mostrado na Figura 50. Caso constate que é uma palavra válida e que condiz com a regra gramatical, o sistema passa, então, para a análise semântica. Para esta etapa, o sistema coleta a informação das classes semânticas armazenadas no atributo especial “ontology”. Com base nas suas classes semânticas, o sistema consulta o *Word Bank* para identificar o *proplet* que representa esta palavra. Com a identificação deste *proplet*, realiza-se outra consulta para retornar os nomes dos *frames* que se relacionam semanticamente com ele. Esta lista é armazenada em memória no componente *Motor* para ser utilizada nas próximas etapas de funcionamento. Para melhor entender, na Figura 51 apresenta-se o exemplo da etapa B.

Para a etapa C, na Figura 52 mostra-se o diagrama de atividades partindo novamente das opções de cores do CS. Após receber a próxima opção selecionada pelo usuário, neste caso a opção “*What Doing?*”, o *Motor* solicita a análise semântica de quais pictogramas se relacionam com a opção “*Who?*”, escolhida anteriormente. Assim, consulta-se a Ontologia

Figura 48 – Diagrama de Atividades da Etapa B



Fonte: do Autor.

Figura 49 – Exemplo de estrutura *template* e filtro de classe gramatical

allo.all **filter.tbl**

```
!template[cat: (n' v), syn: T_1]
![sur, core, ontology]
drink drink (DrinkingAction)
sleep sleep (SleepingAction)
eat eat (EatingAction)
```

table T_1:	[core] => [verb, mdr, arg, nc, pc]
	\1 => \1, (), (), (), ().

table T_2:	[core] => [noun, mdr, fnc, nc, pc]
	\1 => \1, (), (), (), ().

table T_3:	[core] => [adj, mdr, mdd, nc, pc]
	\1 => \1, (), (), (), ().

Fonte: do Autor.

Figura 50 – Exemplo de regras gramaticais do arquivo syntax.rul

```
ST_S = {[cat: _] {MAIN+FV}}      FV+MAIN {FV+MAIN, S+IP}      S+IP {}
MAIN+FV {FV+MAIN, S+IP}
[noun: _, cat: (NP)]
[verb: _, cat: (NP' _ v)]
acopy(SS.noun nw.arg)
acopy(nw.verb SS.fnc)
cancel(NP')
copy(nw)

[verb: _, cat: (NP' _ v)]
[noun: _, cat: (_ NP)]
acopy(SS.verb nw.fnc)
acopy(nw.noun SS.arg)
cancel(NP')
copy(nw)

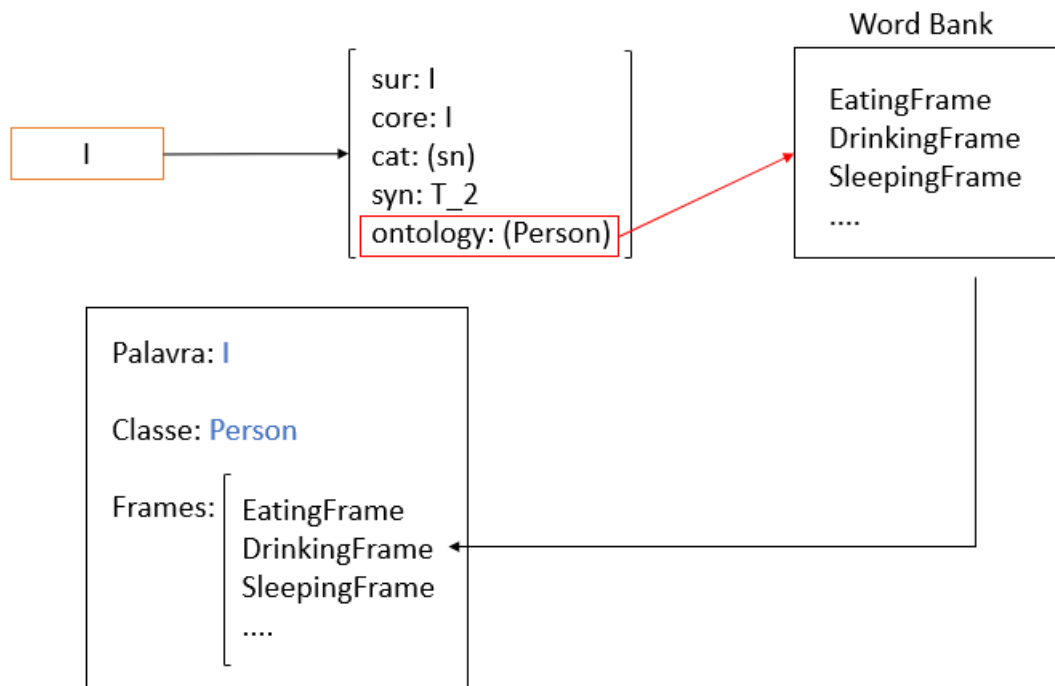
[cat: (v)]
[cat: (v' decl)]
ecopy(decl SS.cat)

ST_F = {[cat: (decl)] rp_S+IP}
```

Fonte: do Autor.

utilizando a lista de nomes dos *frames* armazenados, com base na restrição semântica referente a opção do CS selecionada. Por exemplo, dado que a opção escolhida foi “*What Doing?*” e sabendo que se refere a um verbo, a consulta deve procurar por classes que são declaradas na propriedade semântica “hasMainAction” (cf. Figura 41) de cada *frame* da lista referente ao “*Who?*”. Assim, tem-se como resultado uma lista de classes semânticas utilizadas como parâmetro para encontrar os *proplets* no *Word Bank*. Com os resultados da consulta de *proplets*, a interface do usuário é alterada novamente, apresentado a lista dos novos pictogramas candidatos. Na Figura 53, mostra-se um exemplo de como funciona a etapa C. Ao fim de todo o processo, as próximas ações se iniciam a partir da etapa B apresentado na Figura 45, onde o pictograma é escolhido e as análises são realizadas, sendo neste exemplo para a palavra “*eat*” e posteriormente “*cake*”. É importante destacar que após a seleção do segundo pictograma, referente a etapa B (cf. Figura 48), a estrutura que armazena a lista de *frames* do sujeito é incrementada criando uma segunda lista que representa os *frames* do verbo (cf. Figura 54). Neste sentido, a próxima lista de *frames* a ser utilizada nas consultas semânticas à Ontologia dizem respeito ao verbo. Destaca-se que esta lista relacionada aos verbos deve ser filtrada de modo que considere a classe semântica do sujeito. Por exemplo, na Figura 54, a palavra “*drink*” pode ter uma lista de *frames* que permita que o sujeito seja uma “Pessoa” (e.g., *PersonDrinkingFrame*) ou um “Animal” (e.g., *AnimalDrinkingFrame*). Uma vez selecionado o sujeito com a definição de “Animal”,

Figura 51 – Exemplo da Etapa B



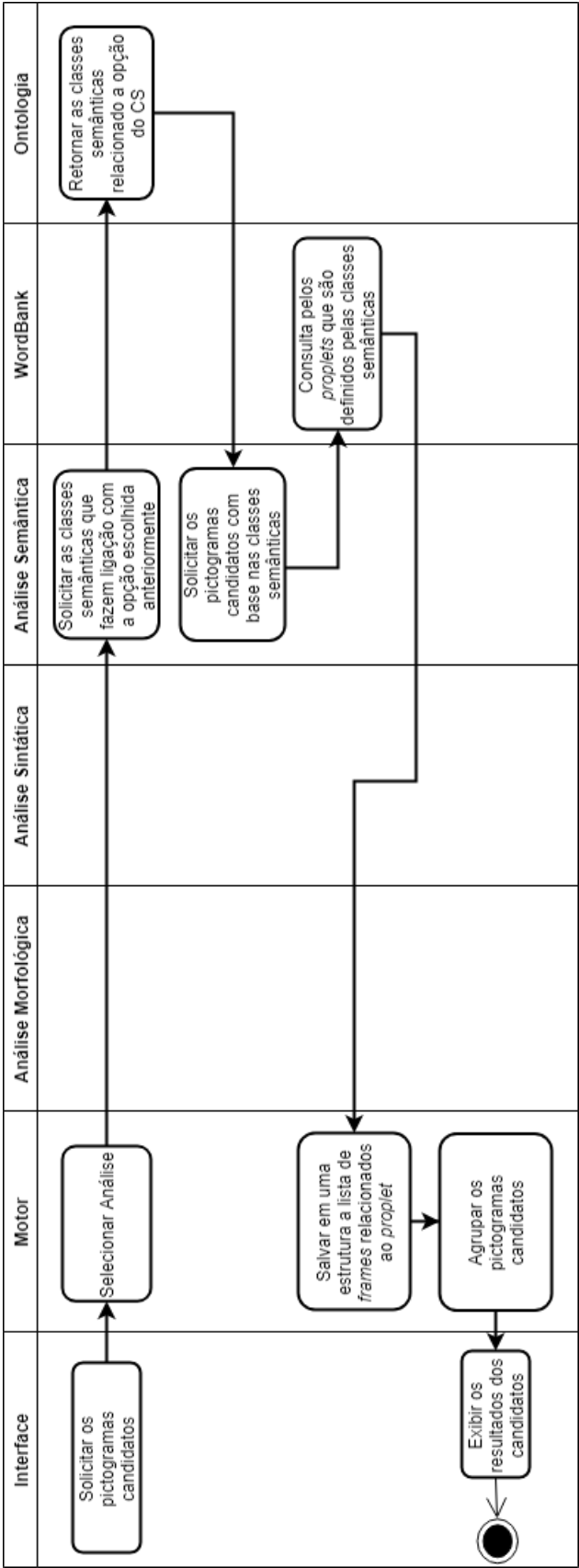
Fonte: do Autor.

o único *frame* que deve ser utilizado como parâmetro para consulta de novos pictogramas candidatos é o “AnimalDrinkingFrame”. Entretanto, esta filtragem não remove da lista o *frame* que não vai ser utilizado, apenas marca o nome com um valor *booleano* “false”, para que não seja usado nas próximas consultas.

O armazenamento de *frames* relacionados ao sujeito e ao verbo flexibiliza a seleção das opções do CS, pois permite que o usuário selecione qualquer opção e, mesmo assim, receba os pictogramas candidatos corretos. Como exemplo, considere que o usuário construiu, até o momento, a frase “*I drink*” e deseja alterar o sujeito da frase. A consulta dos pictogramas candidatos será guiada pelos *frames* semânticos relacionados ao verbo “*drink*”, sem a restrição *booleana* imposta pelo sujeito “*I*” (*Person*) (cf. Figura 54), o que permite pictogramas candidatos do tipo “*Person*” ou “*Animal*”. O inverso também pode acontecer, trocando o verbo “*drink*” por outro que tenha relação semântica com o sujeito. Esta prática também funciona para a alteração das outras opções do CS.

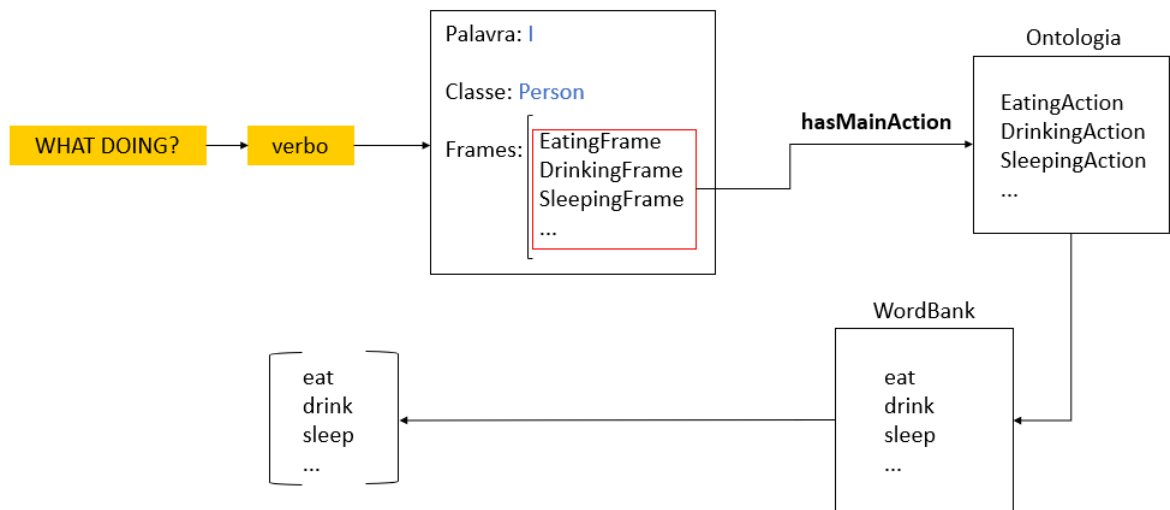
Como resultado final, na Figura 55 apresenta-se a versão do cenário atual em comparação ao exemplo do cenário ideal apresentado na Figura 30 presente na Seção 4.1. Este exemplo está montando a frase “*I drink cold water*”. Como primeira etapa, a interface apresenta as opções do CS, que ao escolher “*Who?*”, é direcionado para a lista de pictogramas candidatos. Ao escolher a opção “*I*”, retorna-se a lista de opções de cores, desta vez, mostrando a primeira palavra escolhida vinculada ao “*Who?*”. Na terceira ação, o usuário escolhe a opção “*What Doing?*”, o que remete a uma lista com verbos. Conside-

Figura 52 – Diagrama de Atividades da Etapa C

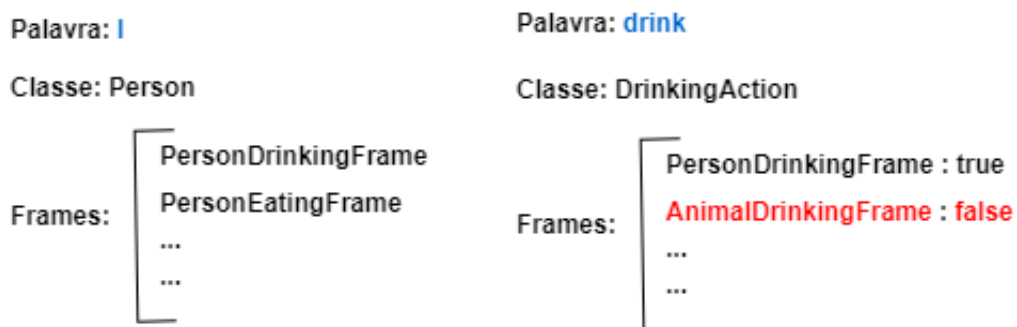


Fonte: do Autor.

Figura 53 – Exemplo da Etapa C



Fonte: do Autor.

Figura 54 – Exemplo das listas de *frames* do verbo restringidos pelo sujeito

Fonte: do Autor.

rando que o usuário tenha escolhido o pictograma candidato “*drink*”, o usuário é então direcionado novamente para a tela de opções de cores. Desta vez, a lista contempla as duas palavras já selecionadas (i.e., “*I*” e “*drink*”) e as demais opções de escolha. Na próxima etapa, o usuário seleciona a opção “*Describe?*”, que remete à adjetivos de algo bebível por uma pessoa, pois o verbo selecionado é “*drink*” e o sujeito é uma pessoa. A escolha do usuário é o candidato “*cold*”. Em seguida, mostra-se a lista de opções do CS novamente, agora com três palavras escolhidas: “*I*”, “*drink*” e “*cold*”. Tendo como objetivo a escolha de algo bebível, o usuário seleciona a opção “*What?*”, este retorna uma lista com líquidos ingeridos por pessoas. Assim, resta ao usuário a escolha do líquido que ele deseja. Neste exemplo, tem-se a palavra “*water*”. Por fim, basta o usuário seleciona a opção “*Talk*” para o sistema vocalizar a frase.

Figura 55 – Exemplo do cenário atual

Choose an option (use the number): 0) Exit 1) Describe? 2) Who? 3) What Doing? 4) Describe? 5) What? 6) Describe? 7) Where? 8) Talk Option: ①	Choose a candidate (use the number): 1) I 2) You 3) Mommy 4) Daddy 5) Sister 6) Dog 7) Cat 8) Bird Option: ②
Choose an option (use the number): 0) Exit 1) Describe? 2) Who? (I) 3) What Doing? 4) Describe? 5) What? 6) Describe? 7) Where? 8) Talk Option: ③	Choose a candidate (use the number): 1) eat 2) drink 3) sleep 4) go 5) run 6) play Option: ④
Choose an option (use the number): 0) Exit 1) Describe? 2) Who? (I) 3) What Doing? (drink) 4) Describe? 5) What? 6) Describe? 7) Where? 8) Talk Option: ⑤	Choose a candidate (use the number): 1) sweet 2) salty 3) hot 4) cold Option: ⑥
Choose an option (use the number): 0) Exit 1) Describe? 2) Who? (I) 3) What Doing? (drink) 4) Describe? (cold) 5) What? 6) Describe? 7) Where? 8) Talk Option: ⑦	Choose a candidate (use the number): 1) water 2) coffee 3) milk 4) tea 5) juice 6) soda Option: ⑧

4.5 AVALIAÇÃO PRÁTICA DO PROTÓTIPO

Segundo (NEWELL; LANGER; HICKEY, 1998; CORNISH; HIGGINBOTHAM, 2000), um SCAA deve: (1) dar poder de expressão aos usuários – i.e., disponibilizar um vocabulário baseado em símbolos e meios para a construção de frases; (2) melhorar a taxa de comunicação – i.e., diminuir o tempo e a quantidade de cliques necessários para se construir uma frase; e (3) facilitar comunicação – i.e., utilizar recursos visuais e de interação que sejam simples para o usuário. Com relação ao poder de expressão e visando as necessidades básicas do usuário de SCAA, o *parser* semântico aqui apresentado faz uso de uma gramática simplificada (i.e., os cinco níveis do CS) que permite a construção de 10 tipos diferentes de frases (cf. Tabela 6). Os 10 tipos de frases abrangem desde construções simples, com 3 constituintes, até construções mais complexas, com 7 constituintes.

Tabela 6 – Comparativo entre as frases geradas e a variação do CS

FRASES	VARIAÇÕES
<i>I drink water</i>	Who - What Doing - What
<i>I drink water house</i>	Who - What Doing - What - Where
<i>happy Daddy drink water</i>	Describe - Who - What Doing - What
<i>Daddy drink bad beverage</i>	Who - What Doing - Describe - What
<i>Daddy drink soda park</i>	Who - What Doing - What - Describe - Where
<i>happy Daddy drink bad beverage</i>	Describe - Who - What Doing - Describe - What
<i>Daddy drink bad beverage big park</i>	Who - What Doing - Describe - What - Describe - Where
<i>happy Daddy drink bad beverage park</i>	Describe - Who - What Doing - Describe - What - Where
<i>happy Daddy drink soda big park</i>	Describe - Who - What Doing - What - Describe - Where
<i>happy Daddy drink bad beverage big park</i>	Describe - Who - What Doing - Describe - What - Describe - Where

Fonte: do Autor.

Dado que o *parser* ainda não foi implementado em um SCAA (e.g., aBoard), este não pode ser testado em um ambiente real com usuários reais. Assim, o modo texto do *parser* é comparado com a versão atual do aBoard. Para isto, utiliza-se como métrica de comparação a quantidade de cliques necessários para criar frases. Neste sentido, baseando-se nos testes realizados em outros trabalhos (GARAY-VITORIA; ABASCAL, 2006; NEWELL; LANGER; HICKEY, 1998; LI; HIRST, 2005; TRNKA; MCCOY, 2007; CARLBERGER et al., 1997) que avaliam a redução de cliques em predição de palavras, adaptou-se a fórmula para o contexto de um *parser* semântico para CAA sem predição de palavras.

Assim, para esta análise, foram testadas todas as possíveis variações de frases (cf. Tabela 6) em relação a quantidade de cliques, utilizando a Equação 4.1 (TRNKA; MCCOY, 2008) para identificar a porcentagem de redução de cliques. Nesta equação, lê-se “KS” como *keystroke savings* (em português, redução de cliques), *keys_ aboard* como a quantidade de cliques no aBoard e *keys_parser* como a quantidades de cliques do *parser* semântico. Ressalta-se que o aBoard possui uma área de conteúdo paginada, ou seja, o usuário precisa fazer cliques ou movimentos de deslizar para acessar os pictogramas das

próximas páginas. Assim, uma vez que o conteúdo do *parser* é apresentado em uma lista não paginada, não se considera a ação de troca de página no aBoard. Em outras palavras, para esta avaliação, considera-se que o pictograma desejado está sempre na primeira página. No caso do aBoard são computados os cliques que representam a ação de escolher a categoria/pasta desejada, selecionar o pictograma, voltar para a tela inicial (i.e., clicar na opção “Início” do *breadcrumb*, utilizado para navegação) e também clicar em “Talk/Falar”. Ademais, para este teste considera-se que não houve erros de seleção durante o percurso para escolha do pictograma no aBoard, isto é, o usuário seguiu o menor caminho para chegar aos pictogramas desejados, sem errar a categoria, tampouco o pictograma. Para contabilizar os cliques do protótipo, levou-se em consideração que a ação de escrever o número desejado e apertar “Enter” equivale a um clique.

Na Tabela 7 apresenta-se o resultado desta comparação. Diante destes resultados, pode-se perceber que, com o uso do *parser*, houve uma redução média de 31,09% na quantidade de cliques em comparação com o uso do aBoard. Esta redução de cliques aumenta conforme o número de elementos envolvidos na frase aumenta. Por exemplo, comparando a frase F1 com a frase F2, têm-se o aumento de 3 cliques para o aBoard e 2 cliques para o *parser*. Dado que neste teste foram utilizadas frases que envolvem somente a ação de “beber”, o aumento constante de 3 cliques no aBoard é decorrente da localização dos pictogramas relacionados ao verbo. Isto poderia ser diferente caso a escolha do verbo fosse outra (e.g., *eat*, *sleep*, *run*). Diante disto, o aumento de cliques para o aBoard é relativo a localização do pictograma desejado pois, quanto mais especializadas forem as categorias, mais cliques serão necessários para chegar ao objetivo. Na Figura 56, apresenta-se um exemplo da hierarquia existente nas categorias, desde a categoria genérica até a apresentação dos pictogramas. A primeira categoria é chamada de “*Nourishment*”, esta se divide nas subcategorias “*Beverage*” e “*Food*”. Na subcategoria “*Beverage*” encontram-se pictogramas relacionados a bebida, enquanto que, na subcategoria “*Food*” encontram-se pictogramas relacionados a comida. Esta relação pode aumentar, dado que é possível especializar, ainda mais, as categorias de “*Beverage*” e “*Food*”. Para a especialização de “*Beverage*”, pode-se criar novas subcategorias que separam bebidas alcoólicas das não alcoólicas, enquanto que, “*Food*”, pode ser dividida em alimentos do tipo vegetal e outra do tipo animal, por exemplo. Para o *parser*, o nível de especialização das categorias não interfere na quantidade de cliques, uma vez que em uma tela o usuário precisa escolher a cor e em outra tela os pictogramas semanticamente relacionados. Por fim, apesar de serem diferenças pequenas na quantidade de cliques, estas são significativas durante o uso real de uma criança da educação especial pois, tornam a comunicação mais rápida.

$$KS = \frac{keys_{aboard} - keys_{parser}}{keys_{aboard}} \times 100\% \quad (4.1)$$

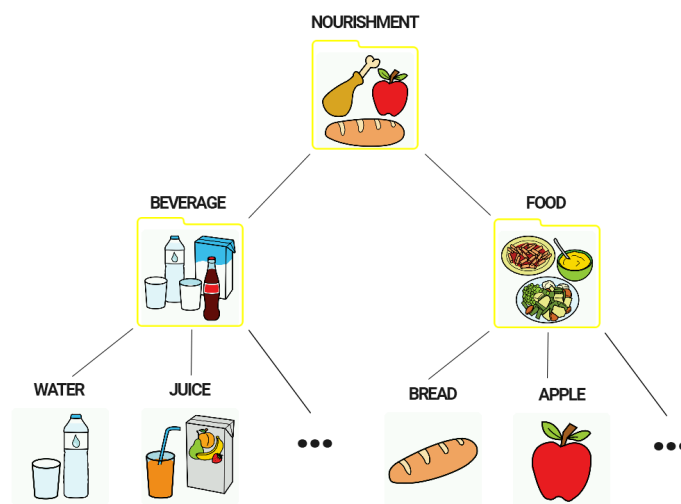
Com relação à facilidade de comunicação, a ausência de uma gramática visual (e.g., estágio atual do aBoard) e de uma interface simples pode interferir na quantidade de

Tabela 7 – Comparativo entre a quantidade de cliques do aBoard e da proposta

#	Frase	aBoard	Proposta	Redução	KS
1	<i>I drink water</i>	10	7	3	30%
2	<i>I drink water house</i>	13	9	4	30,77%
3	<i>happy Daddy drink water</i>	13	9	4	30,77%
4	<i>Daddy drink bad beverage</i>	13	9	4	30,77%
5	<i>Daddy drink soda park</i>	13	9	4	30,77%
6	<i>happy Daddy drink bad beverage</i>	16	11	5	31,25%
7	<i>Daddy drink bad beverage big park</i>	19	13	6	31,58%
8	<i>happy Daddy drink bad beverage park</i>	19	13	6	31,58%
9	<i>happy Daddy drink soda big park</i>	19	13	6	31,58%
10	<i>happy Daddy drink bad beverage big park</i>	22	15	7	31,81%

Fonte: do Autor.

Figura 56 – Especialização da categoria “Nourishment” no aBoard



Fonte: do Autor.

toques/cliques, durante a construção de uma frase, uma vez que o usuário pode divagar na utilização da interface. Em outras palavras, o usuário pode errar a categoria/pasta desejada, bem como se deparar com pictogramas que não tenham relação semântica com a frase, o que pode induzir o usuário ao erro e à criação de frases sem significado. Para exemplificar este cenário, na Figura 57 apresenta-se a criação da frase “*I drink water*” na versão atual do aBoard. Neste exemplo observa-se que nas telas (5) e (6) aparecem pictogramas que não possuem relação semântica com o pictograma “*drink*”. Para contornar este problema, o *parser* utiliza a gramática visual do CS combinado com a interface que alterna entre as telas de opções e a de pictogramas candidatos, guiando o usuário e facilitando o processo de comunicação.

Figura 57 – Criação da frase “I drink water” no aBoard



Fonte: do Autor.

4.6 AVALIAÇÃO COMPARATIVA DO PROTÓTIPO

Nesta seção avalia-se o *parser* semântico em comparação aos trabalhos relacionados no que tange à presença (+) e ausência (-) dos critérios elencados na Seção 3.1. Na Tabela 8 apresenta-se a análise comparativa dos trabalhos relacionados com esta proposta. Com relação ao critério C1 (Léxico Personalizado), a presente proposta utiliza um gerador de léxico personalizado que possibilita a utilização de um vocabulário personalizado para cada usuário. Com isto, independente da forma com que o mediador especifica o vocabulário, este estará disponível para utilização ao usuário sem afetar em nenhum aspecto a funcionalidade do *parser* semântico. Neste sentido, o poder de expressão do usuário vai depender do seu vocabulário personalizado. Para atender ao critério C2 (Gramática Visual), este trabalho utiliza o CS como uma gramática visual para orientar o usuário na criação das frases. Assim, o usuário pode realizar a associação de cores com o significado e a ordem dos pictogramas, o que pode melhorar o aprendizado do usuário na língua e reduzir erros gramaticais. Para atender ao critério C3 (Sugestão de Candidatos), esta proposta utiliza uma interface que se modifica conforme a interação do usuário, apresentando pictogramas candidatos de acordo com restrições impostas pelos pictogramas previamente selecionados pelo usuário. Estes pictogramas candidatos são apresentados após a seleção de uma cor do CS. No que se refere ao critério C4 (Implementação de Ferramenta), a presente proposta especifica e implementa um protótipo que simula o cenário ideal de uma solução real. Tornando a replicação e o teste desta ferramenta disponível para validação. Por fim, para atender o critério C5 (Utilização de *Word Bank*), este trabalho propõe uma forma de armazenar informações do léxico e relações semânticas em um *Word Bank*, fazendo uma mescla com o conhecimento da Ontologia. Com isto, a implementação desta solução em um aplicativo de uso real consome menos recursos de processamento, uma vez que a maior parte deste processamento ocorre no *Word Bank* e não somente em uma Ontologia.

Em suma, o critério C1 (Léxico Personalizado) é abrangido por esta proposta, pelos trabalhos de MARTÍNEZ-SANTIAGO et al. e de NETZER; ELHADAD. No critério C2 (Gramática Visual), apenas esta proposta e o trabalho de PAHISA-SOLé abordam o aspecto de um guia visual. Para a sugestão de pictogramas candidatos (C3), o trabalho de PAHISA-SOLé não apresentou uma proposta que permita sugerir pictogramas candidatos. Por mais que os trabalhos relacionados mencionem sobre uma interface para utilização, apenas esta proposta mostrou a implementação detalhada e tornou disponível para *download*³ a ferramenta (critério C4). No que se refere a utilização de um *Word Bank* (critério C5), somente este trabalho e a proposta de PAHISA-SOLé fazem uso deste recurso.

³ <<https://github.com/augustolima/aboard-semantic-parser>>

Tabela 8 – Análise Comparativa entre os trabalhos relacionados e a proposta

	C1	C2	C3	C4	C5
Martínez-Santiago (2016)	+	-	+	-	-
Netzer (2006)	+	-	+	-	-
Pahisa-Solé (2012)	-	+	-	-	+
Lima (2018)	+	+	+	+	+

Fonte: do Autor.

4.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou uma avaliação que leva em consideração três funcionalidades que um SCAA deve atender: poder de expressão, melhoria na taxa de comunicação e facilidade de comunicação. Neste sentido, para avaliar o poder de expressão, realizou-se a avaliação das diferentes formas de se construir uma frase com base nos cinco níveis do CS. Como resultado, obteve-se 10 formas diferentes de construir frases. No que se refere à melhoria na taxa de comunicação, comparou-se a quantidade de cliques necessários para construir duas frases no aBoard e na proposta. Como resultado constatou-se que a quantidade de cliques no aBoard está relacionado a organização do vocabulário do usuário. Quanto mais especializado, mais cliques serão necessários para construir determinadas frases no aBoard. Isto não acontece na proposta, uma vez que esta proposta possui somente duas telas: seleção da cor do CS e seleção dos pictogramas. Neste sentido, a proposta mostrou uma redução média de 31,09% cliques por frase. Por fim, para avaliar a facilidade de comunicação, realizou-se um comparativo na forma com que o aBoard e esta proposta fazem para construir as frases. Neste sentido, a avaliação considerou a construção da frase “*I drink water*” nas duas abordagens. Na abordagem atual do aBoard, constatou-se que durante a interação o usuário pode encontrar pictogramas que não tenham relação semântica com outros pictogramas já selecionados, o que induz o usuário a erros. Diante disso, a abordagem atual traz facilidade à comunicação, pois com o *parser* semântico é possível apresentar apenas pictogramas candidatos com relação semântica aos pictogramas anteriores.

Dado que o protótipo apresentado simula a utilização de um SCAA em um ambiente de utilização com usuários reais e que o *parser* é apenas uma simulação de uso real, não foi possível realizar testes com o público alvo. Em um ambiente de uso, seria possível avaliar, de forma comparativa, o tempo de construção de uma frase com o aBoard atual e com o aBoard acrescido do *parser*. Além disso, uma avaliação com os usuários permitiria identificar se a interface proposta atende, de forma eficaz, as necessidades do usuário.

Por utilizar apenas cinco níveis do CS, não são considerados os pictogramas de tempo (e.g., *yesterday*, *tomorrow*, *morning*), advérbios (e.g., *quickly*, *sadly*, *gently*) e negação (e.g., *never*, *not*). Isto deve ser considerado em uma futura extensão da gramática, pois

estes pictogramas aumentam o poder de expressão do usuário, permitindo criar frases mais complexas a medida em que seu conhecimento linguístico aumenta. Ainda com relação ao poder de expressão, a interface apresentada não considera a possibilidade de escolher mais de um sujeito e um verbo (e.g., “*Mommy Daddy drink water*”, “*I want drink soda*”), limitando a frase em apenas um pictograma por opção.

5 CONCLUSÃO

Neste capítulo apresentam-se as considerações finais sobre o trabalho desenvolvido, suas principais contribuições, limitações e sugestões de trabalhos futuros.

5.1 CONSIDERAÇÕES FINAIS

A principal motivação deste trabalho baseia-se na dificuldade de expressão e de aprendizado da língua por crianças com deficiência cognitiva e/ou na fala. Devido a estas dificuldades, tais crianças tendem a criar frases incompletas e sem sentido, omitindo verbos e elementos gramaticais, assim como estruturando-as incorretamente. Por este motivo, o SCAA atrelado a um *parser* semântico é uma alternativa com potencial para resolver este problema. As soluções atuais deixam a construção da frase livre, permitindo que frases sem significado sejam construídas. Além disso, dado que estas soluções organizam seu conteúdo por categorias/pastas, a construção de uma frase se torna demorada, uma vez que navegar pela especialização das classes demanda uma quantidade desnecessária de cliques em categorias até encontrar o pictograma desejado. Neste percurso o usuário pode se deparar com categorias/pastas e outros pictogramas que não tenham relação semântica com outro já selecionado, induzindo o usuário a cometer erros na construção da frase ou da escolha de uma categoria.

Neste contexto, este trabalho propôs um *parser* semântico para SCAA que guie, de forma visual e em tempo de execução, a criação de frases telegráficas bem formadas e em ordem direta, bem como sugira pictogramas candidatos (i.e., pictogramas sintática e semanticamente relacionados). Este desenvolvimento foi orientado por requisitos elencados no cenário ideal, os quais refletem à interação de um usuário de CAA no aplicativo aBoard (cf. Seção 4.1). Diante dos requisitos, foi desenvolvido um protótipo composto por um Gerador de Léxico Personalizado e um *Parser* Semântico. O Gerador de Léxico Personalizado tem a responsabilidade de pré-processar informações referentes ao léxico e ao conhecimento semântico dos mesmos. O *Parser* Semântico utiliza as informações do primeiro componente e é responsável por obter, em tempo de execução, os pictogramas candidatos garantindo que a frase esteja bem formada.

A solução proposta apresentou uma nova interface para o usuário de SCAA interagir e se comunicar. Esta visa solucionar os problemas de criação de frases sem significado, reduzir a quantidade de cliques utilizando uma gramática visual que facilita a associação e a ordenação dos pictogramas na frase. Ademais, o uso do *parser* semântico elimina a etapa de percorrer várias categorias/pastas para encontrar um pictograma, visto que a interface se modifica apresentando ao usuário somente os pictogramas candidatos que tenham relação semântica com o pictograma previamente selecionado. Para isto, utilizou-

se o padrão de cores do CS em combinação com o sistema JSLIM adaptado para o contexto de CAA e uma ontologia com conhecimento linguístico e de domínio. As cores do CS possuem grande expressividade com relação a ordem para estruturar a frase e as funções sintáticas de cada cor. Desta maneira, o CS serviu de parâmetro para o funcionamento geral de todo o sistema, guiando o usuário na criação das frases. O CS possibilitou a associação de cores com as diferentes funções sintáticas e papéis semânticos e, serviu como guia de navegação no *Word Bank*. Para o JSLIM, utilizou-se os componentes de análise morfológica e sintática, atribuindo atributos especiais que fazem referência à Ontologia. Para a parte semântica utilizou-se do conceito do DBS para criar o *Word Bank*. Ressalta-se que DBS foi escolhido por fazer uso de um *Word Bank* (i.e., banco de dados), o que melhora o tempo de processamento das análises sintáticas e semânticas e, torna viável a implementação em um dispositivo móvel. Em outras palavras, a utilização de um *Word Bank* para realizar a maior parte da análise sintática e semântica com consultas simples, torna viável sua implementação, devido a dispositivos móveis de baixo custo não possuírem um grande poder de processamento. Neste sentido, a estrutura do *Word Bank* foi adaptada de modo a reduzir o detalhamento dos relacionamentos das palavras. Com esta alteração foi necessário modificar a forma de navegação do DBS e a forma de montar uma frase. Diante da abstração das relações sintáticas e semânticas dos dados armazenados no *Word Bank*, a navegação passou a ser guiada pelo relacionamento com os *frames* semânticos existentes na Ontologia e pela cor do CS. A forma com que a frase deve ser organizada passou a ser orientada pelas cores do CS, sendo então definida na ordem direta. Conforme estas mudanças, a interface passou a se alternar conforme a interação do usuário, de modo que seja escolhido uma cor e em seguida exibido uma lista de pictogramas candidatos que tenham relação semântica com o anterior.

A avaliação deste trabalho foi guiada de modo a verificar três funcionalidades principais que um SCAA deve atender: poder de expressão, melhoria na taxa de comunicação e facilidade de comunicação. Com relação ao poder de expressão, constatou-se que os cinco níveis do CS permitem a criação de 10 tipos de frases. Para a avaliação da taxa de comunicação, comparou-se a construção desses 10 tipos de frases em dois ambientes: o protótipo do *parser* e o aplicativo aBoard. Com os resultados desta comparação, observou-se que com o protótipo houve uma redução média de 4,9 cliques em relação a abordagem atual do aplicativo aBoard. Nesta mesma avaliação, foi observado que a quantidade de cliques no aBoard pode variar dependendo do nível de especialização dos conceitos que agrupam os pictogramas. Com relação a avaliação da facilidade de comunicação, constatou-se que a abordagem proposta facilitou a interação do usuário com a interface, pois esta se resume a somente duas telas, uma para seleção de uma opção do CS e outra para a escolha do pictograma candidatos desejado. Com isso, não há possibilidade do usuário errar a construção de uma frase, dado que sempre serão apresentados pictogramas que resultem em uma frase bem formada. Ao suprir as três funcionalidades de um SCAA, constatou-se que

os critérios da avaliação conceitual do protótipo foram totalmente atendidos.

Por fim, apresenta-se as respostas das Perguntas de Pesquisa (PP) mostradas na Seção 1.2:

- **PP1** “*Como organizar a interface gráfica de um SCAA e a sua forma de interação de forma a guiar o usuário na criação de frases bem formadas e com sentido?*” – a solução proposta oferece uma interface que guia visualmente o usuário por meio do padrão de cores do CS. Este padrão, por ser considerado uma gramática visual, induz o usuário a criar uma frase bem formada e com sentido, uma vez que utiliza das cores para definir a ordem da frase e os papéis sintáticos e semânticos. Assim, têm-se uma interface que alterna entre as opções de cores e os pictogramas candidatos;
- **PP2** “*Como especificar um Parser Semântico para SCAA que facilite a criação de frases telegráficas bem formadas e com sentido?*” – esta proposta detalhou a implementação de um *parser* semântico partindo de uma visão geral, seguindo pela definição da arquitetura e por fim uma explicação detalhada do funcionamento de cada componente. No que se refere ao funcionamento do *parser*, a ordem com que as análises ocorrem dependem diretamente da interação do usuário com a interface, a qual oferece duas telas: seleção da cor desejada e seleção do pictograma candidato;
- **PP** “*O uso de um Parser Semântico aliado a um método de interação para SCAA pode guiar a criação de frases telegráficas bem formadas e com sentido, bem como facilitar a comunicação do usuário em termos de aumento da taxa de comunicação e redução do esforço físico e cognitivo?*” – A partir dos resultados obtidos na avaliação conceitual e comparativa, pode-se perceber que houve uma redução na quantidade de toques/cliques e facilitou-se a construção das frases com uma interface simples.

5.2 CONTRIBUIÇÕES

Após a conclusão deste trabalho, podem ser citadas como contribuições científicas:

- Adaptação da Teoria SLIM: Esta contribuição acrescentou à teoria o conceito de *parser* semântico. Para isto, foi necessário enriquecer a teoria com novos recursos. Estes recursos são definidos por: Colorful Semantics e *Frames* Semânticos. O primeiro foi inserido de modo que o *Internal Matching* utilizado pelo *Surface Compositionality* e *time-Linearity* se baseiem nos padrões gramaticais impostos pelo CS. O segundo, *frames* semânticos, acrescentam o conhecimento semântico de um domínio específico (e.g., CAA) ao conceito de *proplets*. Neste sentido, além da estrutura que define as ligações sintáticas, existe também a ligação semântica definida por estes *frames*. A partir desta mudança foi possível adaptar a teoria de modo que atende-se o objetivo;

- Utilização da Teoria SLIM em outros domínios: A inclusão dos *frames* semânticos permite que qualquer *parser* semântico possa ser implementado utilizando a teoria. Desta forma, qualquer idioma pode ser atendido, desde que as ligações semânticas estejam corretas.

Além das contribuições científicas, também podem ser citadas como contribuições tecnológicas:

- Adaptação do DBS para generalizar os dados do *Word Bank*. Nesta adaptação constatou-se que as informações contidas no *Word Bank* apresentam redundância de informação, armazenando as relações sintáticas e semânticas de cada palavra com outra. Por este motivo, apresentou-se uma abordagem que generaliza estas informações sintáticas e semânticas de modo que não ocorra redundância nos dados armazenados no *Word Bank*;
- Adaptação da forma de navegação no *Word Bank*. Com esta adaptação, foi possível obter os pictogramas candidatos apenas com a generalização da informação, sem a necessidade de percorrer uma grande quantidade de dados armazenados e sem a complexidade de uma gramática extra;
- Uma interface simples para interação com o usuário. Esta contribuição tem relação com a taxa de comunicação e a facilidade de utilização de um sistema SCAA. Esta interface apresenta duas telas diferentes, a primeira diz respeito a escolha de uma cor do CS e a segunda apresenta os pictogramas candidatos para o usuário escolher durante a construção da frase. Com isto, a taxa de comunicação é melhorada, sendo necessário apenas 2 cliques para cada interação. Além disso, a facilidade na comunicação é atendida, dado que o usuário não precisa acessar inúmeras categorias/pastas para chegar ao seu objetivo;
- Implementação de um *parser* semântico no contexto de CAA. Com base nas adaptações realizadas no DBS combinado à uma Ontologia e a utilização de uma nova interface de interação do usuário, criou-se um *parser* semântico para CAA que permite guiar, de forma visual e em tempo real, a criação de frases telegráficas bem formadas e em ordem direta. Além disso, este *parser* é capaz de criar sugestões de pictogramas candidatos que tenham relação sintática e semântica com os demais já selecionados pelo usuário;
- Implementação do *parser* semântico em outro contexto. A ideia de abstrair as relações semânticas para conceitos abstratos (i.e., utilizar a classe semântica e o *frame* semântico da Ontologia para relacionar as palavras semanticamente) reduziu a redundância de informações sintáticas e semânticas do *Word Bank*. Neste sentido, utilizar as relações semânticas contidas em uma Ontologia pode ser implementada

no conceito original do DBS. Além disto, a implementação de um *parser* semântico fora do contexto da CAA se torna viável, uma vez que sejam feitas adaptações na forma com que a validação semântica é realizada. Por exemplo, nesta proposta, o usuário possui uma interface dinâmica que alterna em duas telas: seleção da opção do CS e seleção do pictograma candidatos. Diante disto, seria necessário mudar este conceito de modo que o novo *parser* semântico receba um *input* e para cada palavra verifique no *Word Bank* e na Ontologia se estas possuem relação semântica válida.

5.3 LIMITAÇÕES

Foram identificadas algumas limitações na solução proposta, a saber:

- A solução apresenta um protótipo que simula a utilização de um SCAA real. Devido ao fato do aplicativo aBoard estar em fase de reestruturação para comportar esta solução, não foi possível realizar testes com usuários reais;
- A solução não engloba uma forma de escolher pictogramas de tempo (e.g., *yesterday*, *tomorrow*, *morning*), advérbios (e.g., *quickly*, *sadly*, *gently*) e negação (e.g., *never*, *not*). Por este motivo, apesar do poder de expressão dos cinco níveis do CS, este ainda pode ser melhorado com a inclusão de outros pictogramas;
- Impossibilidade de ter mais de um pictograma para o sujeito e verbo. Esta limitação ocorre devido ao estilo da interface oferecida nesta proposta, que limita somente a uma opção. Isto pode ser contornado posteriormente com uma melhoria na gramática e na interface;
- Restrito à língua inglesa. Por se tratar de uma solução para o idioma inglês, torna-se inviável a utilização desta solução no mercado local brasileiro. Entretanto, para contornar este problema, será preciso criar uma gramática para o idioma desejado.

5.4 TRABALHOS FUTUROS

Trabalhos futuros que podem evoluir ou estender a proposta apresentada nesta dissertação:

- Ampliação da gramática visual. Ampliar a gramática visual de modo que seja possível criar frases com pictogramas de tempo, advérbio e negação. Isto aumenta o poder de expressão do usuário, oferecendo novas formas de construir as frases, possibilitando uma comunicação mais complexa conforme seu conhecimento linguístico aumenta;
- Utilização de alomorfos. Utilizar o pré-processador de alomorfos oferecidos pelo JS-LIM abre procedência para a criação de um sistema COMPANSION. Neste sentido,

o usuário construirá frases telegráficas e como retorno audível terá uma frase completa, incluindo flexão às palavras e incrementando com palavras omitidas (e.g., determinantes, artigos). Utilizar este sistema poderá melhorar a comunicação do usuário, uma vez que as frases estarão completas, bem formadas e na ordem correta;

- Inserção de mais de um pictograma por tipo. Permitir que o usuário consiga adicionar mais de um sujeito e um verbo à frase telegráfica. Como resultado, o usuário poderá construir frases mais complexas;
- Utilização de um preditor semântico. Utilizar um preditor semântico que realize o ranqueamento dos resultados de modo a apresentar os pictogramas candidatos frequentemente utilizados pelo usuário. Com este preditor será possível facilitar ainda mais a comunicação do usuário, de modo que os primeiros pictogramas apresentados para seleção serão os mais frequentemente utilizados;
- Armazenamento de frases. Realizar o armazenamento de frases que o usuário já criou, de modo que economize cliques, melhorando ainda mais a taxa de comunicação e a facilidade de utilização.

REFERÊNCIAS

- AHO, A. V.; ULLMAN, J. D. *Principles of Compiler Design (Addison-Wesley series in computer science and information processing)*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1977.
- AKERKAR, R.; JOSHI, M. Natural language interface using shallow parsing. *IJCSA*, v. 5, n. 3, p. 70–90, 2008.
- ANGELOV, K. Incremental parsing with parallel multiple context-free grammars. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: teste*. [S.l.: s.n.].
- ASHA. *American Speech-Language-Hearing Association*. 2017. <<http://www.asha.org/>>. [Online; acessado 15-01-2018].
- Assistive. *Assistive: Comunicação Alternativa*. 2017. <<http://assistive.cin.ufpe.br/>>. [Online; acessado 15-01-2018].
- BAKER, C. F.; FILLMORE, C. J.; LOWE, J. B. The berkeley framenet project. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. [S.l.], 1998. p. 86–90.
- BERSCH, R. Introdução à tecnologia assistiva. *Porto Alegre: CEDI*, p. 21, 2008.
- BLISS, C. K. *Semantography (Blissymbolics): A Logical Writing for an illogical World*. [S.l.]: Semantography Blissymbolics Publ, 1965.
- BOLDERSON, S.; DOSANJH, C.; MILLIGAN, C.; PRING, T.; CHIAT, S. Colourful semantics: A clinical investigation. *Child Language Teaching and Therapy*, v. 27, n. 3, p. 344–353, 2011. ISSN 0265-6590.
- BORST, W. N. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Tese (Doutorado) — University of Twente, Netherlands, 1997.
- Brasil. *Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. Comitê de Ajudas Técnicas. Tecnologia Assistiva*. Brasília: CORDE, 2009. 138 p.
- BRYAN, A. Colourful Semantics: Thematic Role Therapy. In: *Language Disorders in Children and Adults*. Whurr Publishers Ltd, 2003. p. 143–161. ISBN 9780470699157. Disponível em: <<http://dx.doi.org/10.1002/9780470699157.ch10>>.
- CARLBERGER, A.; CARLBERGER, J.; MAGNUSON, T.; HUNNICUTT, M. S.; PALAZUELOS-CAGIGAS, S. E.; NAVARRO, S. A. Profet, a new generation of word prediction: An evaluation study. *Natural language processing for communication aids*, 1997.
- CHOWDHURY, G. G. Natural language processing. *Annual review of information science and technology*, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003.
- CORNISH, J.; HIGGINBOTHAM, D. J. Aac device testing. *Buffalo, NY: University of Buffalo*, 2000.

- DAHLGREN, K. A linguistic ontology. *International journal of human-computer studies*, Elsevier, v. 43, n. 5-6, p. 809–818, 1995.
- FITZGERALD, E. *Straight Language for the Deaf. A System of Instruction for Deaf Children*. [S.l.]: Alexander Graham Bell Association for the Deaf, 1949. 104 p. ISBN 9780882000763.
- FRANCO, N.; LIMA, T.; LIMA, A.; SILVA, E.; LIMA, R.; CAVALCANTE, T.; FIDALGO, R. aboard: Uma plataforma para educação inclusiva a partir de comunicação aumentativa e/ou alternativa. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2017. v. 28, n. 1, p. 977.
- FRANCO, N. d. M.; LIMA, A. L. de; LIMA, T. P.; SILVA, E. A. da; LIMA, R. J. de; FIDALGO, R. do N. A recall analysis of core word lists over children's utterances for augmentative and alternative communication. In: IEEE. *Computer-Based Medical Systems (CBMS), 2017 IEEE 30th International Symposium on*. [S.l.], 2017. p. 278–283.
- GARAY-VITORIA, N.; ABASCAL, J. Text prediction systems: a survey. *Universal Access in the Information Society*, Springer, v. 4, n. 3, p. 188–203, 2006.
- GRISHMAN, R.; STERLING, J. Analyzing telegraphic messages. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the workshop on Speech and Natural Language*. [S.l.], 1989. p. 204–208.
- GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge acquisition*, Elsevier, v. 5, n. 2, p. 199–220, 1993.
- GUARINO, N. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. [S.l.]: IOS press, 1998. v. 46.
- HANDL, J.; KABASHI, B.; PROISL, T.; WEBER, C. Jslim—computational morphology in the framework of the slim theory of language. In: SPRINGER. *International Workshop on Systems and Frameworks for Computational Morphology*. [S.l.], 2009. p. 10–27.
- HAUSSER, R. *Left-associative Grammar and the Parser NEWCAT*. [S.l.]: CSLI, 1985.
- HAUSSER, R. *Principles of computational morphology*. [S.l.], 1989.
- HAUSSER, R. Complexity in left-associative grammar. *Theoretical Computer Science*, Elsevier, v. 106, n. 2, p. 283–308, 1992.
- HAUSSER, R. *Foundations of computational linguistics*. [S.l.]: Springer, 1999.
- HAUSSER, R. Database semantics for natural language. *Artificial Intelligence*, Elsevier, v. 130, n. 1, p. 27–74, 2001.
- HAUSSER, R. Database Semantics for Natural Language. *Artificial Intelligence (AIJ)*, Elsevier, Dordrecht, v. 130, 2001.
- HAUSSER, R. *Foundations of Computational Linguistics: Man-Machine Communication in Natural Language*. Springer Berlin Heidelberg, 2013. ISBN 9783662039205. Disponível em: <<https://books.google.com.br/books?id=69aoCAAAQBAJ>>.

- HAUSSER, R. et al. A theory of signs for database semantics. *Information Modelling and Knowledge Bases XV*, IOS Press, v. 15, p. 220, 2004.
- HAUSSER, R. R. *A computational model of natural language communication: Interpretation, inference, and production in database semantics*. [S.l.]: Springer Science & Business Media, 2006.
- HORRIDGE, M.; DRUMMOND, N.; GOODWIN, J.; RECTOR, A. L.; STEVENS, R.; WANG, H. The manchester owl syntax. In: *OWLed*. [S.l.: s.n.], 2006. v. 216.
- Integrated Treatment Services. *Colourful Semantics Training*. 2017. <<http://integratedtreatmentservices.co.uk/our-clients/schools-academies/colourful-semantics-training/>>. [Online; acessado 13-01-2018].
- JURAFSKY, D.; MARTIN, J. H. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. [S.l.]: Pearson/Prentice Hall, 2009.
- KIPPER, K.; DANG, H. T.; PALMER, M. et al. Class-based construction of a verb lexicon. *AAAI/IAAI*, v. 691, p. 696, 2000.
- KYCIA, A. *Implementierung der Datenbanksemantik für die natürlichsprachliche Kommunikation*. [S.l.: s.n.], 2004.
- LÁNYI C S E BACSA, E. e. M. R. e. K. Z. e. P. I. The design question of development of multimedia educational software for aphasia patients. *Lect. Notes Comput. Sci.*, v. 3118, p. 6–13, 2004. ISSN 03029743.
- LEVIN, B. *English verb classes and alternations: A preliminary investigation*. [S.l.]: University of Chicago press, 1993.
- LI, J.; HIRST, G. Semantic knowledge in word completion. In: ACM. *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. [S.l.], 2005. p. 121–128.
- LIM, H. A. *Developmental speech-language training through music for children with autism spectrum disorders: Theory and clinical application*. [S.l.]: Jessica Kingsley Publishers, 2011.
- LIMA, T.; SILVA, E.; LIMA, A.; FRANCO, N.; FIDALGO, R. aboard: uma plataforma computacional na nuvem para comunicação alternativa e educação inclusiva. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2017. v. 6, n. 1, p. 102.
- MARTÍNEZ-SANTIAGO, F.; CUMBRERAS, M. À. G.; RÁEZ, A. M.; GALIANO, M. C. D. Pictogrammar: an aac device based on a semantic grammar. In: *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. [S.l.: s.n.], 2016. p. 142–150.
- MARTÍNEZ-SANTIAGO, F.; DÍAZ-GALIANO, M. C.; UREÑA-LÓPEZ, L.; MITKOV, R. A semantic grammar for beginning communicators. *Knowledge-Based Systems*, Elsevier, v. 86, p. 158–172, 2015.
- Mayer-Johnson. *BoardMaker Software*. 2017. <https://www.boardmakeronline.com/>.

- MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, ACM, v. 38, n. 11, p. 39–41, 1995.
- MOUSINHO RENATA E SCHMID, E. e. P. J. e. L. L. e. M. L. e. N. V. Aquisição e desenvolvimento da linguagem : dificuldades que podem surgir neste percurso. *Revista de Psicopedagogia*, v. 25, n. 78, p. 297–306, 2008. ISSN 0103-8486.
- NETZER, Y. *Semantic Authoring for Blissymbols Augmented Communication Using Multilingual Text Generation*. Tese (Doutorado) — Ben-Gurion University of the Negev, Israel, 2006b.
- NETZER, Y.; ELHADAD, M. Using semantic authoring for blissymbols communication boards. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. [S.l.], 2006a. p. 105–108.
- NEWELL, A.; LANGER, S.; HICKEY, M. The rôle of natural language processing in alternative and augmentative communication. *Natural Language Engineering*, Cambridge Univ Press, v. 4, n. 1, p. 1–16, 1998.
- PAHISA-SOLé, J. *Compansion system for a pictogram-based AAC application in Catalan*. 2012.
- PALAO, S. *ARASAAC - Portal Aragonês de Comunicação Aumentativa e Alternativa*. 2017. <<http://catedu.es/arasaac/>>. [Online; acessado 10-01-2018].
- PENNINGTON, C. A.; MCCOY, K. F. Providing intelligent language feedback for augmentative communication users. In: *Assistive Technology and Artificial Intelligence*. [S.l.]: Springer, 1998. p. 59–72.
- RANTA, A. *Grammatical framework: Programming with multilingual grammars*. [S.l.]: CSLI Publications, Center for the Study of Language and Information, 2011.
- Saltillo. *Saltillo: Low-tech Communication Board Options*. 2017. <<https://saltillo.com/chatcorner/content/29>>. [Online; acessado 15-01-2018].
- SPEARS, C. L.; TURNER, V. L. *Rising to new heights of communication and learning for children with autism: The definitive guide to using alternative-augmentative communication, visual strategies, and learning supports at home and school*. [S.l.]: Jessica Kingsley Publishers, 2010.
- STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: principles and methods. *Data & knowledge engineering*, Elsevier, v. 25, n. 1-2, p. 161–197, 1998.
- TOMASELLO, M. *Constructing a language: A usage-based theory of language acquisition*. [S.l.]: Harvard university press, 2009.
- TRNKA, K.; MCCOY, K. F. Corpus studies in word prediction. In: ACM. *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*. [S.l.], 2007. p. 195–202.
- TRNKA, K.; MCCOY, K. F. Evaluating word prediction: framing keystroke savings. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. [S.l.], 2008. p. 261–264.