

Geração Automática de Sentenças em Língua Natural para Sequências de Pictogramas como Apoio à Comunicação Alternativa e Ampliada

Automatic generation of natural language sentences for pictogram sequences in support of Augmentative and Alternative Communication

Rafael Pereira
Universidade Federal de Sergipe
rafaelps@dcomp.ufs.br

Hendrik Macedo
Universidade Federal de Sergipe
hendrik@dcomp.ufs.br

Rosana Givigi
Universidade Federal de Sergipe
rosanagivigi@uol.com.br

Marco Túlio Chella
Universidade Federal de Sergipe
marco@dcomp.ufs.br

Resumo

A Comunicação Alternativa e Ampliada (CAA) é uma área de prática clínica educacional para fonoaudiólogos cujo objetivo é auxiliar indivíduos que possuam deficiência na oralidade. Os símbolos de comunicação pictórica constituem um dos sistemas da CAA que podem complementar ou mesmo substituir a linguagem falada desses indivíduos. É possível utilizar a habilidade já adquirida em comunicação pictórica por parte de crianças com deficiência para promover sua alfabetização. Infelizmente, a literatura relacionada parece não indicar solução prática para tal questão. Neste artigo, propomos um método para geração automática de sentenças naturais em língua portuguesa do Brasil que corresponda a uma dada sequência de símbolos pictóricos apresentados. Este método foi implementado em uma ferramenta visual de apoio ao profissional educador e atualmente faz parte de um dos recursos de CAA do Laboratório de CAA da Universidade Federal de Sergipe. Um conjunto de validação fornecido pelo Laboratório mostrou a correteza das sentenças geradas pela ferramenta.

Palavras chave

Geração de Linguagem Natural, Comunicação Alternativa e Ampliada, Símbolos Pictóricos

Abstract

The Augmentative and Alternative Communication (AAC) is an area of clinical educational practice for speech therapists whose goal is to assist individuals who are orally deficient. The pictorial communication symbols are one of the AAC systems that can complement or even replace the spoken language of these individuals. It is possible to use the ability already

acquired in pictorial communication by children with disabilities to promote their literacy. Unfortunately, the related literature does not seem to indicate a practical solution to this question. In this paper, we propose a method for automatic generation of natural sentences in the Brazilian Portuguese language in regards to a given sequence of pictorial symbols presented. This method has been implemented in a visual tool to support professional educators and is currently part of one of the AAC tools of the AAC Laboratory at the Federal University of Sergipe, Brazil. A validation set provided by the Laboratory has shown the correctness of the sentences generated by the tool.

Keywords

Natural Language Generation, Augmented Alternative Communication, Pictograph Symbols

1 Introdução

Tecnologia Assistiva é o termo empregado a todo conjunto de dispositivos utilizados para auxiliar indivíduos com algum tipo de limitação intelectual, motora, visual ou auditiva a realizar atividades a que normalmente não estariam completamente aptos (Bharucha et al., 2009; Brodwin, 2010).

Um uso particular das tecnologias assistivas é feito pela chamada Comunicação Alternativa e Ampliada (CAA) (Beukelman & Mirenda, 2005; Alant & Bornman, 1994; Light, 1989). A CAA é uma área de prática clínica de pesquisa e educacional para fonoaudiólogos que visa auxiliar indivíduos que demonstrem prejuízos nos modos de comunicação gestual, oral e/ou escrita.



DOI: 10.21814/lm.9.1.242

This work is Licensed under a

Creative Commons Attribution 4.0 License

Os sistemas de CAA dividem-se em picturais e linguísticos. Dentre os picturais destacam-se o Picture Communication Symbols (PCS), o Pictogram-Ideogram Communication (PIC), o Picsyms, o Rebus e o ARASAAC¹

O sistema de símbolos de comunicação pictórica pode substituir ou complementar a linguagem falada e, desta forma, contribuir para o aumento da interação comunicativa dos indivíduos com deficiência na oralidade, suprimindo as necessidades de recepção, compreensão e expressão da linguagem. Quando se utiliza o computador para CAA, o sistema de símbolos pictóricos associado a um mecanismo de entrada apresenta símbolos representativos que são selecionados pelo usuário, compondo uma mensagem que pode ser estruturada em um texto com apresentação na tela, sintetizado em voz ou a combinação de ambos.

Ainda são escassas as soluções de software e hardware para CAA para uso em computadores convencionais. Grande parte das propostas estão relacionadas à confecção de hardware específicos, tais como teclados e mouses especiais, que possuem alto custo, grandes dimensões e exigem grande treinamento para que todo potencial seja usufruído (Stephanick et al., 2010; Salsman et al., 2010). Estas características dificultam sobremaneira sua disseminação e uso por parte de laboratórios de informática de escolas convencionais. Uma iniciativa acadêmica recente alinha a confecção de um dispositivo de entrada do tipo mordedor com dois diferentes softwares: um para promover aceleração e corretude linguística da redação através da previsão inteligente de palavras e orações futuras e outro para gerar a sequência correspondente de símbolos pictóricos para uma dada sentença redigida em português do Brasil (Santos et al., 2015).

Ainda não existe, entretanto, solução para uma demanda essencial e que corresponde exatamente ao oposto da citada: como gerar automaticamente uma sentença em linguagem natural a partir de uma dada sequência de símbolos pictóricos? Solução apropriada para esta questão seria uma importante ferramenta de apoio à alfabetização de crianças com paralisia cerebral. Além disso, esta mesma solução poderia ser instrumento de comunicação efetivo para crianças já familiarizadas com a comunicação via símbolos pictóricos,

Dois trabalhos são parcialmente relacionados à problemática. Em YAG (McRoy et al., 2000), a solução combina a abordagem baseada em *tem-*

plate (*template-based*) para representar a estrutura de texto com método para representação de conhecimento. Um *template* é uma forma predefinida contendo slots que são então preenchidos com informações especificadas por usuários. O texto gerado pelo YAG pode advir de diferentes tipos de entradas, como uma sequência de proposições em linguagem lógica ou uma estrutura de características junto com o nome do *template*. A aplicação desenvolvida por Ramos-Soto et al. (2015) gera pequenos termos de previsões meteorológicas a partir de desenhos relacionados, como chuva, sol, nuvens, representadas em forma de dados numéricos. A solução consiste na combinação de técnicas de percepção, computação com palavras (Zadeh, 2002, 1996) e estratégias para descrição linguística de dados.

Neste artigo, propomos um método para solução do problema, que consiste fundamentalmente em um modelo baseado em *templates* para Geração de Linguagem Natural, similar ao proposto por McRoy et al. (2000, 2003). Este método foi implementado sob a forma de um software de apoio ao profissional da área de fonoaudiologia ou educação especializada no trabalho de alfabetização de crianças que se utilizam desses símbolos pictóricos para comunicação.

O método para geração automática de sentenças representativas das sequências de símbolos pictóricos é apresentado na seção 2 deste artigo. A ferramenta desenvolvida a partir desta proposta é apresentada na seção 3, onde fazemos preliminarmente a análise de corretude de sentenças geradas para um conjunto de validação fornecido. A seção 4 traz a conclusão do artigo.

2 Método

O método proposto segue um pipeline de ações para geração de texto em linguagem natural linguisticamente correto e que traduza fielmente a semântica da sequência de símbolos pictóricos apresentada como entrada.

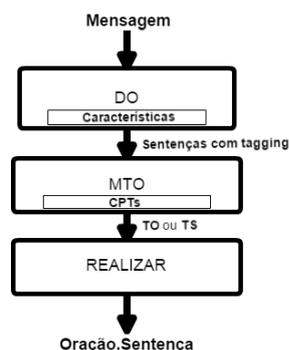


Figura 1: Componentes do método.

¹Clik Tecnologia Assistiva, disponível em http://www.clik.com.br/clik_01.html.

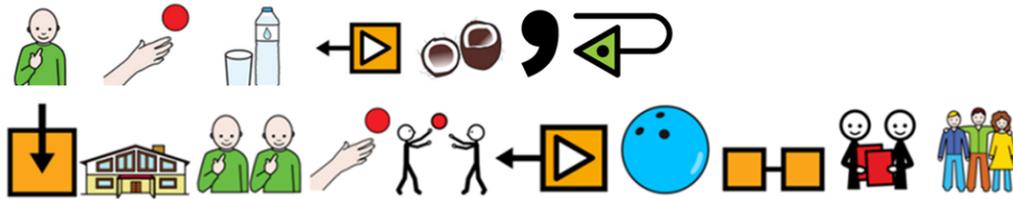


Figura 2: Sequência de símbolos pictóricos (ou mensagem M) de entrada do pipeline de geração.

A figura 2 ilustra um exemplo de sequência de símbolos pictóricos para a qual o método deve produzir como saída de processamento a seguinte sentença, composta de duas orações: *Eu quero beber água de coco, mas em casa nós queremos brincar de bola com nossos amigos.*

O método para solução deste problema é fundamentado na fusão das ideias da gramática gerativa (GG) (Chomsky, 1965) e na representação de conhecimento (RC) através de *templates* (McRoy et al., 2000, 2003; Reiter, 1995). Da GG, a relevância para este problema reside na base categorial que faz parte do componente sintático, na qual uma oração é formada pelo *SN + SV*, onde *SN* é um sintagma nominal e *SV* um sintagma verbal (para este método o *SV* não inclui o verbo). Da RC, utilizamos, em nível de abstração, o conceito dos sintagmas nominal e verbal que foram representados na forma de *Proposição de Template* (PT). Uma *proposição* é uma parte de uma oração, podendo ser um sujeito ou predicado da mesma. Dessa forma, uma PT é um micro-template que pode representar o sintagma *SN* ou *SV* de uma oração. Com isso, qualquer que seja o nível de granularidade de um *template*, ele deve possuir a estrutura sintática característica da língua portuguesa do Brasil.

Para gerar orações ou sentenças por meio de *templates*, o método deve realizar as seguintes tarefas: (i) compor *Template de Oração* (*TO* ~ *SN + SV*) através de combinação de PTs com alguma *Att* (que é o verbo da oração); (2) compor *Template de Sentença* (*TS*) através de combinação de *TOs*, caso a mensagem de entrada necessite; e (3) realizar linguisticamente *TOs* ou *TSs*. Tanto *TO* quanto *TS* devem estar em conformidade com a mensagem (M) que foi dada como entrada para o CTO. Todas essas tarefas são, respectivamente, atribuídas aos módulos *DO*, *CTO* e *Realizar*, que são apresentados na figura 1.

A figura 1 apresenta a ligação entre os módulos que compõem este método. Os módulos são: *Detector de Oração* (*DO*), que por sua vez contém um conjunto de características que identificam uma oração em uma sentença, *Construtor de Template de Oração* (*CTO*), que contém um

conjunto de PTs, e *Realizar*, cuja função é fazer a realização linguística de um template. Mais detalhes sobre esses módulos serão descritos nas próximas subseções.

Na subseção 2.1, apresentamos o procedimento para aquisição do conhecimento e sua representação na base de *templates*. A subseção 2.2 descreve a técnica para detectar e separar as orações de uma sentença. A subseção 2.3 descreve o planejamento de documento e micropLANEJAMENTO. Por fim, a subseção 2.4 descreve como é feita a realização linguística dos *templates*.

Aquisição e Representação do Conhecimento

A criação e validação do *corpus* linguístico, o qual foi utilizado para extrair o conhecimento necessário para este método, foram realizadas com supervisão de pesquisadora-chefe e estudantes de fonoaudiologia do Departamento de Fonoaudiologia da Universidade Federal de Sergipe.

As orações e sentenças que fazem parte desse *corpus*, quando relacionadas com as sequências de símbolos que as representam, possuem os seguintes níveis cognitivos: iconicidade, sintaxe e memorização. A iconicidade consiste na compreensão e percepção, ao selecionar pictogramas que represente alguma oração ou sentença. A sintaxe consiste no uso de preposição, pronome, advérbio e pontuação. Quanto à oração que se deseja gerar, ela pode ser simples ou composta. Já a memorização está relacionada à quantidade de símbolos pictóricos que podem ser representados por uma oração ou sentença. Além disso, os níveis cognitivos tercem o domínio deste método, juntamente com a necessidade de produzir orações que expressem noção de ação ou estado. Estes níveis são também aplicados às sentenças apresentadas na tabela 2.

Assim, a partir da análise desse *corpus*, foram confeccionadas 128 PTs que compõem a base de conhecimento para a solução do problema. Uma PT é então representada por *slots* (que são indicados por <SLT>, <SLV> ou <ATPN>), palavras da língua portuguesa e pela *Att*. A *Att* é in-

dicada pelo símbolo <ATPN>, onde *A* significa a atitude, *T* o tempo verbal, *P* a pessoa relacionada ao verbo (que é o sujeito da oração) e *N* o numeral (que pode ser singular ou plural) e está relacionado ao verbo. Dessa forma, cada PT deve pertencer a um *Conjunto de Proposição de Template* (CPT). Um CPT consiste no agrupamento de PTs por sua *chave*, a qual referencia o mesmo. Isso só é possível porque cada pictograma possui um ou mais nomes que o representam.

Segue alguns exemplos de PTs, CPTs, TO e TS que foram montados a partir de PTs:

1. PTs:

```
&a <SLT_NN>; &ao <SLT_NN>; &com &muita <SLV_VB>
&de <SLT_NN>; &neste <SLT_NN> &de <SLT_NN>
```

2. CPTs:

(a) Chave = IN_NN_PRP

(i) em <SLT_NN> <SLT_PRP>;

anzóis=[em]

(b) Chave= VB_IN_NN_IN_PRP\$ _NN

(i) <SLV_VB> de <SLT_NN> com nossas <SLT_NN>;

anzóis=[de, com, nossas]

(ii) <SLV_VB> de <SLT_NN> com nossos <SLT_NN>;

anzóis=[de, com, nossos]

3. TOs:

```
<ATPN> &com &muita <SLV_VB> &de <SLT_NN>;
```

```
&a <SLT_NN> <AP3S> &ao <SLT_NN>
```

4. TS:

```
<ATPN> &com &muita <SLV_VB> &de <SLT_NN>, mas
```

```
&neste <SLT_NN> &de <SLT_NN> <AP1S> <SLV_VB>
```

Note que nos *templates* que foram apresentados, introduzimos o conceito de “anzóis”, indicados pelo símbolo &, que são as palavras classificadas morfológicamente como artigos, pronomes (exceto os pessoais), conjunções e preposições. Foi dessa forma, que representamos todo o conhecimento necessário para este método.

Detecção de Oração (DO)

Quando uma sequência de pictogramas representa uma mensagem que é composta por mais de uma oração, faz-se necessário identificar e extrair as orações. Isto acontece porque a estratégia é montar um determinado *template* para representar a estrutura de uma única oração ou sentença. Para tal tarefa, o algoritmo 1 é aplicado sobre uma mensagem *m*:

Considere uma mensagem *m*, como a do exemplo da figura 2, para o algoritmo 1. No passo 2 deste algoritmo, um pré-processamento sobre os nomes dos símbolos é realizado para que sejam retiradas as extensões e quaisquer caracteres que

Algoritmo 1 Detector de oração.

```
1: procedure DETECTAORACAO(m)
2:   m ← PREPROCESSAMENTO(m)
3:   sentencas ← DETECTESENTENCA(m)
4:   for all sentencas do
5:     tagSentencas ← POSTAG(sentencas)
6:   for all tagSentencas do
7:     oracoes ← ←
      REGRAS(sentencas, tagSentencas)
8:   return oracoes
```

não componham uma palavra válida na língua portuguesa (do Brasil). Então temos *m* igual a:

```
eu querer beber água de coco,
mas em casa nós querer brincar
de bola com nossos amigos.
```

No passo 3, detecta-se as sentenças que foram armazenadas em *m*. Em seguida, a função *POSTag* (Jurafsky & Martin, 2000) é aplicada sobre todas as sentenças (*sentencas*) nos passos 4 e 5. Ambas funções utilizadas, nos passos 3 e 5 do algoritmo 1, foram implementadas pela biblioteca OpenNLP.²

Já nos passos 6 e 7, são extraídas as orações através da aplicação das características apresentadas na tabela 1. Dentro da função REGRAS, cada vetor de *tokens* e vetor de morfemas de cada sentença são então varridos a fim de encontrar alguma característica listada na tabela 1. Quando isso acontece, uma oração é detectada e, então, esta é atribuída à variável *oracoes*.

Por fim, no passo 8, as orações que foram detectadas são retornadas. Sendo assim, obtemos como resultado do Detector de Orações para a mensagem *m*:

- i) eu_NNP querer_VB beber_VB água_NN
de_IN coco_NN ,_SYM
- ii) em_IN casa_NN nós_PRP querer_VB
brincar_VB de_IN bola_NN com_IN
nossos_PRP\$ amigos_NN

A Tabela 1 apresenta oito tipos de características que podem ocorrer na estrutura sintática da língua portuguesa, conforme as sentenças apresentadas no *corpus* (veja a subseção 2.1). Essas características são notadas quando se varre um vetor de *token* que representa algum sentença (tokenizada por espaço em branco e sinal de pontuação). Além do mais, para que as características sejam válidas, elas devem atender à pré-condição de que ao menos um verbo

²Disponível em <http://opennlp.apache.org/>.

Id	Morfema do <i>Token</i> Corrente	Morfema do <i>Token</i> Anterior	Morfema do <i>Token</i> Posterior
1	Conjunção	—	—
2	Vírgula	—	—
3	Pronome demonstrativo	Preposição	—
4	Pronome pessoal	—	Verbo
5	Pronome pessoal	Verbo	Verbo
6	Verbo	Vírgula	Vírgula
7	Interrogação	—	—
8	Exclamação	—	—

Tabela 1: As características que identificam uma oração para o nosso escopo.

deve existir antes da posição $i - 1$ do *token* corrente i , exeto a sexta característica.

O Construtor de Templates de Orações (CTO)

O módulo CTO para esta ferramenta foi baseado nos módulos de planeamento definidos por Reiter & Dale (2000). Ele tem a responsabilidade de montar a estrutura sintática de uma oração ou sentença através do TO que pode ser concatenado a fim de montar o *template da sentença* (TS) de acordo com a sequência de pictogramas.

Com o resultado (i) do DO para sequência de pictogramas (ou mensagem **M**) ilustrado na figura 2, temos que as *proposições* são **M1**="eu" e **M2**="beber água de coco", consequentemente, **SN**="PRP", **Att**="querer" e **SV**="VB_NN_IN_NN" (para este módulo da ferramenta consideramos o SN e SV à nível morfológico, respectivamente, das *proposições M1* e **M2**). Desta forma, as *chaves* que mapeiam o conjunto onde devem ser encontradas as PTs de **M1** e **M2** são *chave(M1)*="PRP" e *chave(M2)*="VB_NN_IN_NN". Os resultados dessas *chaves* foram obtidos via concatenação dos valores armazenados no array de morfemas (visto na subseção 2.2). Então, para este exemplo, temos que os CPTs são:

1. Chave(M1) = PRP

(a) <SLT_PRP> anzóis=[]

2. Chave(M2) = VB_NN_IN_NN

(a) <SLV_VB> <SLT_NN> com <SLT_NN>
anzóis=[com]

(b) <SLV_VB> <SLT_NN> de <SLT_NN>
anzóis=[de]

O primeiro *template* do CPT é selecionado, identificando-se pela *chave(M1)*, que representa a *proposição M1*="eu". Este CPT contém apenas PT formado por *slot*, ou seja, qualquer um deles pode ser selecionado. A questão é como selecionar o *template* que melhor representa a

proposição M2="beber água de coco". Para isso, utiliza-se o possível "anzol" de **M2**, &de, de modo que o template escolhido do CPT da *chave(M2)* é o segundo elemento do conjunto. O *template* completo para a oração TO_i seria então:

<SLT_PRP> <Att> <SLV_VB> <SLT_NN> de
<SLT_NN>

Seja a segunda saída (ii) do DO, *em_IN casa_NN nós_PRP querer_VB brincar_VB de_IN bola_NN com_IN nossos_PRP\$ amigos_NN*. As *proposições* neste caso são **M1**="em casa nós", **M2**="brincar de bola com nossos amigos" e **Att**="querer". Assim, **SN**="IN_NN_PRP", **SV**="VB_IN_NN_IN_PRP\$NN". As *chaves* que mapeiam os CPTs, onde deve ser encontrada as PTs **M1** e **M2**, são *chave(M1)*="IN_NN_PRP" e *chave(M2)*="VB_IN_NN_IN_PRP\$NN". Estas *chaves* mapeiam para os seguintes CPTs:

1. Chave(M1) = IN_NN_PRP

(a) em <SLT_NN> <SLT_PRP>
anzóis=[em]

2. Chave(M2) = VB_IN_NN_IN_PRP\$NN

(a) <SLV_VB> de <SLT_NN> com nossas
<SLT_NN>
anzóis=[de, com, nossas]

(b) <SLV_VB> de <SLT_NN> com nossos
<SLT_NN>
anzóis=[de, com, nossos]

Do primeiro conjunto, é selecionado o único template que representa a *proposição M1*="em casa nós". A seguir, o segundo elemento do CPT identificado pela *chave(M2)* é selecionado, já que os possíveis anzóis são &de, &com &nossos e que o template escolhido está relacionado à *proposição M2*="brincar de bola com nossos amigos". O template completo para a oração TO_{ii} é:

em <SLT_NN> <SLT_PRP> <Att> <SLV_VB> de
<SLT_NN> com nossos <SLT_NN>

Caso a *chave(M1)* ou *chave(M2)* não referenciem nenhum CPT deste módulo, será necessário

inserir na base do sistema *templates* (PTs) que representem a oração desejada. Para que ainda assim se tenha ao menos uma sentença como saída, faz-se necessário um procedimento particular: outra *chave* dentre as existentes deve ser selecionada, desde que seja semelhante à *chave*(M1) ou *chave* (M2). Esta semelhança será computada através da similaridade do cosseno (intervalo [0, 1]):

$$\cos(z_i) = \frac{\vec{u} \cdot P_i}{\|\vec{u}\| \|P_i\|},$$

tal que $0 \leq i \leq 9$ e $0 \leq z_i \leq \frac{\pi}{2}$, onde a *chave* de uma determinada proposição é o vetor $\vec{p} = (m^1, m^2, \dots, m^9)$ com m sendo o valor da enumeração morfológica (número de classes gramaticais consideradas) e $P = (\vec{a}, \vec{b}, \dots, \vec{n})$ é o conjunto de vetores pertencentes ao módulo CTO. O vetor com maior valor de similaridade será selecionado.

Sendo assim, o TS da sentença ilustrada na figura 2 é montado ao concatenar TO_i , a conjunção “mas” e $TO_{(ii)}$, então temos que TS é igual a:

<SLT_PRP> <Att> <SLV_VB> <SLT_NN> de <SLT_NN>, mas em <SLT_NN> <SLT_PRP> <Att> <SLV_VB> de <SLT_NN> com nossos <SLT_NN>

Realização Linguística

O propósito da realização linguística é realizar os TOs ou TSs (veja a subseção 2.3), ou seja, preencher os slots com as palavras correspondentes, respeitando a concordância nominal, fazer a concordância verbal correta com o sujeito da oração e, finalmente, adicionar os sinais de pontuação.

Dois dicionários de palavras foram construídos, considerando apenas palavras relacionadas com os símbolos pictóricos presentes na base. O primeiro dicionário possui palavras que não são verbos e está organizado da seguinte forma: (i) a primeira palavra é a palavra-chave que identifica as demais e não está flexionada em gênero ou número, (ii) as próximas palavras são flexionadas por número e, depois, por gênero. O segundo dicionário possui apenas verbos. O primeiro verbo está na forma infinitiva e funciona como a palavra-chave. Os verbos seguintes estão flexionados nos tempos verbais Presente e Futuro para cada pronome. Com este dicionário, pode-se realizar um template nesses dois tempos verbais. Por padrão, o tempo e a pessoa verbal utilizado para realização de *template* é o presente do indicativo e terceira pessoa do singular. Para as demais palavras que não são verbos, o número é singular e o gênero, masculino por padrão.

Para o TO_i do primeiro exemplo, *<SLT_PRP> <Att> <SLV_VB> <SLT_NN> de <SLT_NN>*, a realização linguística inicia com $PT(M1) = <SLT_PRP>$ preenchendo-se o slot com o pronome “eu”; isto resulta em $PT(M1)' = \text{“eu”}$. A seguir *Att = “querer”* deve concordar com o sujeito da *proposição M1* e deve ser conjugado no presente do indicativo (porque nenhum tempo verbal foi informado pelo usuário) de forma a concordar com o pronome: *Att' = “quero”*. Por fim, preenche-se os slots da $PT(M2) = <SLV_VB> <SLT_NN> de <SLT_NN>$ e, dessa forma, temos que $PT(M2)' = \text{“beber água de coco”}$. Ao concatenar $PT(M1)'$, *Att'*, $PT(M2)'$ e realizar a pontuação, temos a oração “eu quero beber água de coco.”.

Para o $TO_{(ii)}$ do segundo exemplo, *em <SLT_NN> <SLT_PRP> <Att> <SLV_VB> de <SLT_NN> com nossos <SLT_NN>*, o primeiro slot do template da $PT(M1) = em <SLT_NN> <SLT_PRP>$ é relacionado com o nome do pictograma “casa” e o segundo slot com o pronome “nós”: $PT(M1)' = \text{“em casa nós”}$. Em seguida, *Att = “querer”* deve concordar com o sujeito da *proposição M1*, que nesse caso é o pronome e deve ser conjugado no presente do indicativo (por *default*): *Att' = “queremos”*. Finalmente, preenche-se os slots da $PT(M2) = <SLV_VB> de <SLT_NN> com nossos <SLT_NN>$ com as descrições dos pictogramas, respectivamente, “brincar”, “bola” e “amigos”: $PT(M2)' = \text{“brincar de bola com nossos amigos”}$. Ao concatenar $PT(M1)'$, *Att'* e $PT(M2)'$, a oração realizada é “em casa nós queremos brincar de bola com nossos amigos”.

Com isso, temos que TS (= $TO_i + TO_{(ii)}$) realidizado linguisticamente é igual a esta sentença “eu quero beber água de coco, mas em casa nós queremos brincar de bola com nossos amigos”.

Discussão

Tendo em vista viabilizar uma comunicação simples e autônoma aos pacientes que ainda estão se familiarizando com os símbolos pictóricos, se faz necessário passar como entrada para o método ao menos um símbolo que represente um verbo (atitude). Isso se dá porque é bastante comum o uso de orações que expressem noção de ação ou estado neste tipo de comunicação.

Pela mesma razão em que se exige uma atitude, o uso de vírgulas na entrada do DO (ver seção 2.2) não é obrigatório. Se a vírgula fosse omitida no exemplo da subseção 2.2 depois da palavra “coco”, ainda assim seria possível detectar a oração, pois neste caso, a entrada casaria

Nível	Descrição das Sentenças
1	Vamos tomar sorvete comigo
1	A menina foi para o mercado, mas não tinha dinheiro
2	A cidade é muito fria, por isso, o homem precisou de dois casacos para não adoecer
2	A festa foi hoje, se não tivesse chovido, ganharia muitos presentes, pois convidei muitos amigos
3	O tempo está chuvoso, por isso, não esqueça de fechar as janelas da casa quando sair para não molhar os móveis
3	Neste fim de semana, fui para fazenda de vovô. Calvaguei, me banhei de rio, comi manga, bebi leite da vaca e brinquei com meus amigos que moram lá

Tabela 2: Exemplos de sentenças fornecidas pelo Departamento de Fonoaudiologia.

com a característica 1 da tabela 1 e estaria em conformidade com esta regra da gramática: use vírgula antes das conjunções “mas”, “porém”, “pois”, “embora”, “contudo”, “todavia”, “portanto” e “logo”.

De certo, na montagem de TO ou TS, existe a amarração de PTs aos *anzóis* quando os utilizam para selecionar um PT de CPT (veja a subseção 2.3), embora isso ocorra somente se for passado algum símbolo pictórico como entrada para o método, que é interpretado como um *anzol*.

3 A Ferramenta CA²JU ESCRITO

O método de geração proposto para conversão de sequência de pictogramas em texto natural foi aplicado no desenvolvimento de uma ferramenta de apoio ao profissional que lida, em particular, com crianças que fazem uso da CAA para se comunicarem.

A composição visual ordenada de símbolos pictóricos deve ser feita da seguinte forma: (1) o profissional seleciona os pictogramas que estão apresentados em um teclado virtual localizado na parte inferior da imagem (figura 3), (2) os símbolos selecionados são apresentados em ordem da seleção no campo acima do teclado e (3) o texto será gerado a partir de um click.

Os símbolos utilizados na ferramenta pertencem ao sistema ARASAAC, que fora desenvolvido pelo Portal Aragonês de CAA. Esta é uma obra de Sergio Palao para CATEDU,³ que os publica sob a licença Creative Commons.

Esta ferramenta faz atualmente parte de um conjunto de recursos de CAA do Laboratório de CAA da Universidade Federal de Sergipe para ensaios clínicos com pacientes.

A escolha de mensagens (em forma de sequência de símbolos pictóricos) que compõe a base experimental para testes e validação com crianças é baseada em protocolos de avaliação bem definidos pelos profissionais do Laboratório

de CAA. Estes protocolos visam a seleção do sistema de signos por meio da compreensão, da percepção visual (escolha dos símbolos, tamanhos, etc), da mobilidade (acesso aos sistemas de auxílio técnico: precisão, rapidez, agilidade, força, etc), do nível cognitivo (nível de iconicidade, memória, léxico), de aspectos linguísticos, das posições posturais (ex: sentado, deitado, etc). Por fim, os protocolos analisam as formas de indicação dos sinais, sendo possível: (i) indicação direta, (ii) direta com auxílio, (iii) codificada, (iv) varredura (ou exploração) dependente ou (v) varredura independente. Um protocolo de acompanhamento vem sendo desenvolvido para registro semanal dos dados que evidenciem como estão sendo atingidos os objetivos comunicativos para a ferramenta.

A corretude dos textos produzidos pela ferramenta foi avaliada comparando-se com um conjunto de validação fornecido pelo Laboratório (ver tabela 2). A distância de Levenshtein, utilizada como métrica neste experimento preliminar, apontou valor próximo de 0 (zero) para todo o conjunto. Isto significa que a similaridade léxico-sintática entre as sentenças geradas automaticamente pela ferramenta e as pertencentes ao conjunto de validação foi muito alta.

4 Conclusão

Este artigo propôs um método para geração automática de sentenças em linguagem natural a partir de sequência de símbolos pictóricos, bastante utilizados em suporte à Comunicação Alternativa e Ampliada (CAA). O método proposto é baseado na confecção de *templates* que permitem boa variabilidade linguística das construções.

O método descrito foi utilizado para criação de uma ferramenta de suporte ao profissional de CAA que lida com crianças com paralisia cerebral e com crianças com transtorno do espectro autístico. O propósito específico da ferramenta é propiciar um ambiente computacional para facilitar a alfabetização destas crianças. A literatura relacionada não mostra quaisquer iniciativas com

³<http://catedu.es/arasaac/>



Figura 3: Interface gráfica Ca2ju Escrito.

este propósito. A ferramenta possui interface visual adequada para composição de sequência de pictogramas por parte do profissional e posterior geração do texto natural correspondente.

Em experimentação preliminar com um conjunto de validação fornecido pelo Laboratório de CAA da Universidade Federal de Sergipe foi mostrado que a geração das sentenças por parte da ferramenta condiz perfeitamente com as sentenças do conjunto de validação. A ferramenta é atualmente integrante do conjunto de recursos de CAA do respectivo laboratório e faz parte dos ensaios clínicos com grupos de controle e experimental.

Trabalhos em andamento consistem no aumento do conjunto de validação e complexidade das sentenças-alvo, finalização do protocolo de acompanhamento e, principalmente, avaliação quantitativa da contribuição da ferramenta enquanto mecanismo da CAA para a alfabetização de crianças com deficiência a partir dos grupos citados anteriormente. Resultados destes estudos são previstos até fim de 2017.

Referências

- Alant, Ema & Juan Bornman. 1994. Augmentative and alternative communication. *South African Family Practise* 15(5).
- Beukelman, David & Pat Mirenda. 2005. *Augmentative and alternative communication*. Brookes Publishin.
- Bharucha, Ashok J., Vivek Anand, Jodi Forlizzi, Mary Amanda Dew, Charles F. Reynolds, Scott Stevens & Howard Wactlar. 2009. Intelligent assistive technology, applications to dementia care: Current capabilities, limitations, and future challenges. *The American Journal of Geriatric Psychiatry* 17.
- Brodwin, Martin. 2010. Assistive technology. Em Irving B. Weiner & W. Edward Craighead (eds.), *Corsini Encyclopedia of Psychology*, 1–2. John Wiley and Sons.
- Chomsky, Noam. 1965. *Aspects of the theory of syntax*. MIT Press.
- Jurafsky, Daniel & James H. Martin. 2000. *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition* Prentice Hall Series in Artificial Intelligence. Prentice Hall.
- Light, Janice. 1989. Toward a definition of communicative competence for individuals using augmentative and alternative communication systems. *Augmentative and Alternative Communication* 5(2). 137–144.
- McRoy, Susan W., Songsak Channarukul & Syed S. Ali. 2000. YAG: a template-based generator for real-time systems. Em *1st International Conference on Natural Language Generation (INLG)*, vol. 14, 264–267.
- McRoy, Susan Weber, Songsak Channarukul & Syed S. Ali. 2003. An augmented template-based approach to text realization. *Natural Language Engineering* 9(4). 381–420.
- Ramos-Soto, Alejandro, Alberto Jose Bugarín, Senén Barro & Juan Taboada. 2015. Linguistic

- descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems* 23(1). 44–57.
- Reiter, Ehud. 1995. NLG vs. templates. Em *European Workshop on Natural Language Generation*, vol. 5, 95–106.
- Reiter, Ehud & Robert Dale. 2000. *Building natural language generation systems* Natural Language Processing. Cambridge University Press.
- Salsman, Kenneth, John Sweetser & Anders Grunnet-Jepsen. 2010. Electronic equipment for handheld vision based absolute pointing system. Patente 7796116. US Patent and Trademark Office.
- Santos, Flávio, Carlos Junior, Hendrik Macedo, Marco Chela, Rosana Givigi & Luciano Barbosa. 2015. CA²JU: an assistive tool for children with cerebral palsy. *Studies in Health Technology and Informatics* 216. 589–593.
- Stephanick, James, Christina James, Ethan R. Bradford & Michael R. Longé. 2010. Selective input system based on tracking of motion parameters of an input device. Patente 7750891. US Patent and Trademark Office.
- Zadeh, Lofti A. 1996. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems* 4(2). 103–111.
- Zadeh, Lofti A. 2002. From computing with numbers to computing with words – from manipulation of measurements to manipulation. *International Journal of Applied Mathematics and Computer Science* 12(3). 307–324.