



Análise Semântica com base em AMR para o Português

AMR-based Semantic Parsing for the Portuguese Language

Rafael Torres Anchiêta  
Instituto Federal do Piauí

Thiago Alexandre Salgueiro Pardo  
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Resumo

A Representação Abstrata de Significado (no inglês, *Abstract Meaning Representation* —AMR) é um formalismo semântico projetado para capturar o significado de uma sentença, representando-a como um grafo direcionado acíclico de única raiz com nós rotulados (conceitos) e arestas (relações) entre os nós. Essa representação tem recebido bastante atenção da comunidade de Processamento de Língua Natural, pois muitos autores têm proposto vários modelos de análise semântica para produzir grafos AMR a partir de uma sentença, visando melhorar o entendimento da língua natural. Entretanto, a maioria desses modelos focam no inglês devido a falta de grandes *corpora* anotados para outras línguas, deixando uma lacuna entre o inglês e outros idiomas. A fim de superar esse problema, neste artigo, é realizada uma análise detalhada de vários analisadores AMR, adaptando três modelos diferentes para o português e propondo melhorias. Além disso, estendeu-se um analisador baseado em regras desenvolvido previamente para o português. Esses modelos foram avaliados sobre um *corpus* anotado para o português. Por fim, realizou-se uma análise detalhada de erros com o objetivo de identificar os maiores desafios para análise no português e obter *insights* que possam ajudar pesquisas futuras nesta área.

Palavras chave

representação abstrata de significado, análise semântica, Português

Abstract

Abstract Meaning Representation (AMR) is a semantic formalism designed to capture the meaning of a sentence, representing it as a single rooted directed acyclic graph with labeled nodes (concepts) and edges (relations) among them. This representation has received growing attention from the Natural Language Processing community as many authors have proposed several models to produce an AMR graph from a sentence, aiming to improve natural language

understanding. However, most of these models have focused on the English language due to the lack of large annotated corpora for other languages, producing a gap between English and other languages. To overcome this issue, in this paper, we carried out a fine-grained analysis of several parsers, adapted three different models to Portuguese, and proposed some improvements. Furthermore, we extended a previous rule-based AMR parser designed for Portuguese. We evaluated these models on a manually annotated corpus in Portuguese. Then, we performed a detailed error analysis to identify the major challenges in Portuguese AMR parsing that we hope will inform future research in this area.

Keywords

abstract meaning representation, semantic parsing, Portuguese

1. Introdução

A semântica computacional é a área encarregada de estudar representações semânticas viáveis computacionalmente para expressões na língua humana (Jurafsky & Martin, 2009). Nesta área, um analisador semântico, também conhecido como *parser* semântico, é responsável por verter o conteúdo de um texto em uma representação semântica computacional de maneira automática. Normalmente, isso ocorre abstraindo fenômenos sintáticos do texto e identificando, por exemplo, os sentidos das palavras, entidades nomeadas, papéis semânticos e outras características semânticas, visando eliminar interpretações ambíguas do texto (Goodman et al., 2016).

O desenvolvimento de analisadores semânticos é motivado pela hipótese de que a semântica pode ser usada para melhorar muitas tarefas de Processamento de Língua Natural (PLN), tais como: sumarização automática (Liu et al., 2015; Hardy & Vlachos, 2018), geração automática de



texto (Pourdamghani et al., 2016; Song et al., 2017, 2018), vinculação de entidades (Pan et al., 2015; Burns et al., 2016), detecção de paráfrase (Issa et al., 2018; Anchieta & Pardo, 2020b), sistemas de perguntas e respostas (Mitra & Baral, 2016) e tradução automática (Song et al., 2019), entre outras, produzindo sistemas melhores e mais informados.

Uma representação semântica é um dos ingredientes mais importantes de um analisador semântico. Por isso, vários pesquisadores têm empregado bastante esforço na criação de representações semânticas, como, por exemplo: a tradicional Lógica de Primeira Ordem, detalhada por Pereira & Shieber (2002) e Jurafsky & Martin (2009), Redes Semânticas (Lehmann, 1992), *Universal Networking Language* (UNL) (Uchida et al., 2006), *Universal Conceptual Cognitive Annotation* (UCCA) (Abend & Rappoport, 2013) e, mais recentemente, *Abstract Meaning Representation* (AMR) (Banarescu et al., 2013).

AMR, em particular, tem ganhado muita atenção da comunidade científica devido a sua estrutura relativamente simples, mostrando relações semânticas entre conceitos através de um grafo direcionado, como exibido na Figura 1.

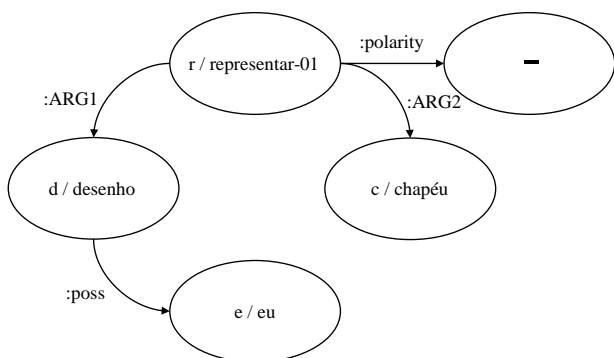


Figura 1: Exemplo de grafo AMR para a sentença “Meu desenho não representava um chapéu.”, extraída do *corpus* composto pelo livro “O Pequeno Príncipe” (Anchieta & Pardo, 2018a).

Na Figura 1, os nós são conceitos e as arestas são as relações semânticas. O conceito *representar-01* é a raiz do grafo e *:ARG1*, *:ARG2*, *:poss* e *:polarity* são relações do formalismo AMR.

A principal motivação para o uso de representações semânticas como AMR se deve ao fato de que elas apresentam a semântica de maneira explícita, permitindo realizar uma análise mais informada do conteúdo textual e dos resultados alcançados. Embora arquiteturas neurais, como as baseadas em *transformers*, tenham/estão atingindo resultados do estado da arte em diversas

tarefas de PLN, o conhecimento semântico utilizado nessas modelagens está implícito em vetores densos (as conhecidas *word embeddings*), tornando a análise e interpretação dos resultados mais difícil. Há várias tarefas de PLN em que a semântica explícita pode ser necessária, como extração de informação, ferramentas de apoio à leitura e escrita e simplificação textual, entre outras. Também é interessante destacar as críticas que a semântica implícita tem recebido (veja, por exemplo, as críticas de Bender & Koller (2020)).

De acordo com Banarescu et al. (2013), a criação do formalismo AMR foi motivada pela necessidade de prover para a comunidade científica *corpora* com anotações semânticas relacionadas às tarefas tradicionais de PLN, tais como: reconhecimento de entidades nomeadas, anotação de papéis semânticos, desambiguação do sentido de palavras, resolução de correferência, e outras. A partir de *corpora* disponíveis, muitos analisadores AMR, de diferentes abordagens, foram desenvolvidos, por exemplo, baseados em grafo (Flanigan et al., 2014), em árvores de dependência (Wang et al., 2015b), em sistemas de transição (Damonte et al., 2017), em aprendizado profundo (van Noord & Bos, 2017; Lyu & Titov, 2018) e em *transformers* (Cai & Lam, 2020), entre outras.

A grande maioria desses analisadores está disponível apenas para o inglês devido à falta de grandes *corpora* anotados em outros idiomas. Essa escassez de recursos produz uma lacuna entre o inglês e outras línguas. Uma alternativa para preencher essa lacuna e criar aplicações mais efetivas de PLN é adaptar analisadores do inglês para outras línguas. Seguindo essa estratégia, Wang et al. (2018) adaptaram o analisador semântico de Wang et al. (2015b) para a língua chinesa, produzindo o primeiro analisador AMR para esse idioma.

Com base nesse contexto, este trabalho apresenta adaptações e melhorias de alguns analisadores AMR do inglês para o português. Além disso, investigou-se como as estratégias adaptadas se desempenham em uma língua com um *corpus* anotado pequeno. Realizou-se uma análise profunda de três analisadores AMR, identificando seus pontos fortes e suas fraquezas. Essa análise possibilitou a implementação de melhorias nesses analisadores. Junto com analisadores adaptados, propôs-se uma melhoria de um analisador desenvolvido previamente para a língua portuguesa.

Com objetivo de avaliar o desempenho dos analisadores, conduziu-se um experimento comparando grafos gerados automaticamente e grafos construídos manualmente, utilizando duas

métricas de avaliação de parsers AMR, Smatch (Cai & Knight, 2013) e SEMA (Anchieta et al., 2019). Por fim, realizou-se uma análise detalhada de erros visando identificar os principais desafios para análise AMR na língua portuguesa, esperando auxiliar e fomentar pesquisas futuras nesta área.

No geral, este artigo faz as seguintes contribuições: (i) adaptação e melhoria de modelos de *parsing* AMR do inglês para o português, (ii) melhoria de um analisador desenvolvido para o português, (iii) uma visão geral dos analisadores AMR e (iv) uma análise detalhada de erros dos analisadores AMR adaptados.

O restante do artigo está organizado da seguinte maneira: a Seção 2 introduz os principais conceitos sobre o formalismo AMR; na Seção 3, são apresentados os *corpora* disponíveis nesta área; a Seção 4 oferece uma visão geral dos principais métodos de análise AMR; na Seção 5, é detalhada a adaptação de analisadores AMR para o português, bem como a implementação de melhorias; a Seção 6 reporta os experimentos conduzidos e os resultados obtidos; por fim, na Seção 7, conclui-se o artigo, indicando-se futuros direcionamentos.

2. Fundamentos de AMR

Abstract Meaning Representation (AMR) é um formalismo semântico produzido com o objetivo de capturar o significado de uma sentença, abstraindo elementos da estrutura sintática, como informação morfossintática e ordem das palavras (Banarescu et al., 2013). Esse formalismo descarta palavras que considera que contribuem pouco para o significado essencial da sentença, como artigos. Além disso, ele foca na estrutura predicado-argumento de uma sentença, conforme definido pelo PropBank (Kingsbury & Palmer, 2002; Palmer et al., 2005).

AMR pode ser representado como um grafo direcionado acíclico de raiz única com nós (conceitos) e arestas (relações) rotuladas. Os nós representam os principais eventos e entidades mencionados em uma sentença, enquanto as arestas representam o relacionamento semântico entre os nós, conforme apresentado na Figura 1. Os conceitos AMR podem ser concretos, compreendendo palavras em sua forma lexicalizada (“mulher”, “homem”), *framesets* do PropBank (“representar-01”) ou abstratos (como palavras chave especiais), que não correspondem a nenhuma unidade lexical da sentença, tais como *email-address-entity*, *percentage-entity*, *distance-quantity*, entre outros.

Além da estrutura em grafo, AMR pode ser representado em outras notações: tradicionalmente, na lógica de primeira ordem, para comparar e avaliar duas estruturas AMR, ou na notação PENMAN (Matthiessen & Bateman, 1991), para facilitar a leitura e a anotação humana. Na Figura 2, é exibido um exemplo de AMR na notação PENMAN (lado esquerdo) e um exemplo na lógica de primeira ordem (lado direito), respectivamente, para o grafo da Figura 1.

<pre>(r / representar-01 :polarity - :ARG1 (d / desenho :poss (e / eu)) :ARG2 (c / chapéu))</pre>	<pre>instance(r, representar-01)^ instance(d, desenho)^ instance(e, eu)^ instance(c, chapéu)^ polarity(r, '-')^ ARG1(r, d)^ poss(d, e)^ ARG2(r, c)</pre>
--------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 2: No lado esquerdo, notação PENMAN para AMR e, no lado direito, lógica de primeira ordem.

Outra característica do formalismo AMR é a natureza do relacionamento (alinhamento) entre as palavras de uma sentença e os nós do grafo. Nesse formalismo, não existe alinhamento explícito entre os nós do grafo e as palavras de uma sentença, ou seja, o alinhamento não faz parte da estrutura do AMR, embora ele seja importante para a tarefa de análise semântica.

Para avaliar estruturas AMR produzidas automaticamente por um *parser* semântico, utiliza-se tradicionalmente a métrica Smatch (Cai & Knight, 2013). Essa métrica computa o grau de sobreposição de uma estrutura AMR automática e uma de referência (produzida por humanos, normalmente), produzindo valores de precisão, cobertura e medida-f. Mais recentemente, Anchieta et al. (2019) desenvolveram uma nova métrica para avaliar estruturas AMR, chamada SEMA. Essa métrica lida com alguns problemas da métrica Smatch (como não considerar a dependência entre elementos e distorcer o cômputo de alguns valores para certas situações de estruturação AMR), sendo mais rápida e robusta do que a Smatch. Além de serem usadas para a tarefa de avaliar analisadores AMR, essas métricas também são úteis para a tarefa de anotação semântica, permitindo, por exemplo, aferir a concordância entre humanos.

Na próxima seção, os principais *corpora* disponíveis na área são apresentados.

Língua	Disponível	Corpus	Treinamento	Desenvolvimento	Teste	Total
Inglês	LDC	LDC2013E117	8.684	1.085	1.085	10.854
		LDC2014T12	10.441	1.305	1.305	13.051
		LDC2015E86	16.833	1.368	1.371	19.572
		LDC2016E25	36.521	1.368	1.371	39.260
		LDC2017T10				
		LDC2020T02	55.635	1.722	1.898	59.255
Inglês	Público	O Pequeno Príncipe	1.274	145	143	1.562
		Bio AMR	5.452	500	500	6.452
Chinês	Público	O Pequeno Príncipe	1.274	145	143	1.562
Português			1.274	145	143	1.562
Espanhol				50		

Tabela 1: *Corpora* AMR

3. Corpora AMR

Existem vários *corpora* disponíveis para o inglês e algumas iniciativas para outras línguas.

Para o inglês, o *Linguistic Data Consortium* (LDC) é o principal responsável por disponibilizar *corpora* manualmente anotados. Os textos desses recursos são de diferentes domínios, como: notícias, fóruns de discussão, blogs e outros. Na Tabela 1, são detalhados os *corpora* disponíveis.

Os *corpora* LDC2015E86, LDC2016E25 e LDC2017T10 possuem as mesmas sentenças para os conjuntos de desenvolvimento e teste, e os *corpora* LDC2016E25 e LDC2017T10 são iguais¹. A comunidade científica utiliza amplamente esses recursos, embora eles não estejam publicamente disponíveis. Até o momento, existem apenas dois recursos anotados públicos para o inglês²: o *corpus* produzido a partir do livro “O Pequeno Príncipe” e o Bio AMR. O primeiro contém o texto completo do livro escrito por Antoine de Saint-Exupéry, publicado em 1943 e traduzido para 300 línguas, enquanto que o segundo inclui textos de domínio biomédico, extraídos da PubMed³.

Para *corpora* de outras línguas, existem algumas iniciativas, tais como em chinês (Li et al., 2016), português (Anchieta & Pardo, 2018a; Sobrevilla Cabezado & Pardo, 2019) e espanhol (Migueles-Abraira et al., 2018). Essas iniciativas anotaram uma versão de “O Pequeno Príncipe” em suas respectivas línguas. Para o espanhol, anotaram-se apenas 50 sentenças. Além dessas iniciativas monolíngues, existe um *cor-*

pus multilíngue, o AMR 2.0–*Four Translations*⁴. Esse *corpus* possui 5.484 sentenças do *corpus* LDC2017T10 traduzidas para o italiano, espanhol, alemão e mandarim, ou seja, 1.371 sentenças para cada língua.

4. Trabalhos relacionados

Diversos estudos foram desenvolvidos para gerar estruturas AMR automaticamente a partir de sentenças. Aqui, são apresentados os principais estudos categorizados em seis classes: baseados em grafo, árvore, sistema de transição, gramática categorial combinatória, aprendizado profundo e *transformers*. Utilizando essas categorias, esta seção é dividida em três subseções: analisadores AMR para o inglês (4.1), analisadores para outras línguas (4.2) e, por último, um resumo dos analisadores (4.3).

4.1. Analisadores para o inglês

4.1.1. Métodos baseados em grafos

Métodos baseados em grafos identificam nós e computam pontuações de arestas para criar ligações entre os nós, adotando o algoritmo *maximum spanning connected subgraph*.

Flanigan et al. (2014) desenvolveram o primeiro analisador para o inglês, chamado JAMR. Os autores abordaram o problema em dois estágios: identificação de conceitos e identificação de relações. No primeiro estágio, eles atacaram o problema como uma tarefa de rotulação de sequência, adotando um modelo de semi-Markov para mapear blocos de palavras em uma sentença para nós no grafo. No estágio de identificar relações, os autores propuseram um algoritmo com o objetivo de encontrar um subgrafo

¹O *corpus* LDC2017T10 está disponível para todos os inscritos no LDC, enquanto o *corpus* LDC2016E25 é limitado aos participantes do DEFT.

²<https://amr.isi.edu/download.html>

³<https://www.ncbi.nlm.nih.gov/pubmed/>

⁴<https://catalog ldc.upenn.edu/LDC2020T07>

fortemente conectado sobre os conceitos identificados no primeiro estágio. Com o objetivo de desenvolver um algoritmo que aprenda esses dois estágios, os autores criaram um alinhador que mapeia as palavras de uma sentença aos nós do grafo correspondente. Com essa abordagem, o método desenvolvido alcançou 58% na métrica Smatch sobre parte do *corpus* LDC2013E117.

Werling et al. (2015) usaram o trabalho de Flanigan et al. (2014) como base e propuseram um novo método para identificação de conceitos, pois 38% das palavras do conjunto de desenvolvimento do *corpus* LDC2013E117 não são vistas durante o treinamento, tornando as abordagens baseadas em memorização frágeis. Os autores avaliaram a abordagem desenvolvida sobre o *corpus* LDC2014T12 e sobre parte do *corpus* LDC2013E117, atingindo 62% e 63,3% na métrica Smatch, respectivamente.

4.1.2. Métodos baseados em árvores

Abordagens baseadas em árvores iniciam a partir de uma árvore de dependência que é incrementalmente modificada para se tornar uma estrutura AMR.

Wang et al. (2015b) criaram um analisador AMR, chamado CAMR, que envolve dois passos. No primeiro passo, o modelo converte uma sentença em uma árvore de dependência, enquanto que o segundo passo transforma a árvore de dependência em um grafo AMR, realizando uma série de ações. Por exemplo, uma dessas ações é transformar a preposição “em” na relação AMR :location. Uma das principais vantagens desta abordagem é o uso de um analisador de dependência que pode ser treinado em um grande *corpus*. O analisador CAMR obteve 63% na métrica Smatch sobre parte do *corpus* LDC2013E117. Em um trabalho posterior, Wang et al. (2015a) adicionaram uma nova ação para inferir conceitos abstratos. Além disso, eles incorporaram características mais ricas produzidas por analisadores auxiliares, tais como: anotador de papéis semânticos e resolvidor de correferência. Os autores reportaram uma melhoria de 7% na métrica Smatch.

Goodman et al. (2016) melhoraram o analisador proposto por Wang et al. (2015b), aplicando algoritmos de aprendizagem por imitação (Osa et al., 2018), visando reduzir ruídos produzidos pelo analisador CAMR. Com essa estratégia, os autores atingiram uma performance similar ao trabalho de Wang et al. (2015a) em parte do *corpus* LDC2013E117.

4.1.3. Métodos baseados em sistemas de transição

Um sistema de transição é uma máquina abstrata caracterizada por um conjunto de configurações (pilha de palavras parcialmente processadas, um *buffer* com palavras não vistas) e transições.

Zhou et al. (2016) propuseram um sistema de transição com o objetivo de aliviar a propagação de erros nos métodos baseados em grafo, executando conjuntamente tarefas de identificação de conceitos e relações em um modelo incremental. O modelo dos autores alcançou 67% na métrica Smatch sobre o *corpus* LDC2014T12.

Damonte et al. (2017) introduziram um analisador inspirado pelo sistema de transição ArcEager de Nivre (2004). A principal diferença entre eles é que o primeiro considera o mapeamento entre palavras de uma sentença e nós AMR, a não projetividade das estruturas AMR e nós de re-entrada (múltiplas arestas de entrada). Projetividade está relacionada à condição de não cruzamento de arestas, como apresentado na Figura 3. Com essa estratégia, o método atingiu 64% na métrica Smatch sobre o *corpus* LDC2014T12.

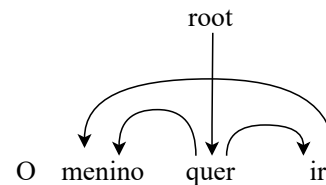


Figura 3: Um exemplo de não projetividade para a sentença “O menino quer ir”. A aresta conectando a palavra “ir” para a palavra “menino” cruza outra aresta.

Ballesteros & Al-Onaizan (2017) utilizaram uma pilha de redes recorrentes para representar o estado do analisador. Além disso, eles adotaram algoritmos gulosos para tomar decisões sobre cada transição. Essa estratégia foi avaliada em parte do *corpus* LDC2013E117 e no *corpus* LDC2014T12, alcançando 69% e 64% na métrica Smatch, respectivamente.

Peng et al. (2018) apresentaram um sistema de transição que generaliza técnicas de análise de dependência com o objetivo de produzir grafos AMR. Para isso, eles usaram um *cache* de tamanho fixo, permitindo que o sistema crie arestas, ao mesmo tempo, para cada vértice no *cache*. Os autores avaliaram esse método no *corpus* LDC2015E86, atingindo 65% na métrica Smatch.

4.1.4. Métodos baseados em Gramática Categorial Combinatória

Gramática Categorial Combinatória (GCC) é um formalismo que provê uma interface transparente entre sintaxe e semântica (Steedman, 1996, 2001).

Artzi et al. (2015) mapearam sentenças para estruturas AMR em um processo de dois estágios. No primeiro, os autores adotaram um sistema de GCC para construir representações de cálculo lambda para aspectos composicionais do AMR. No segundo, eles propuseram um algoritmo de gramática de indução GCC para produzir grafos AMR a partir das representações da primeira etapa. Esse método atingiu 66,1% na métrica Smatch sobre parte do *corpus* LDC2013E117.

Misra & Artzi (2016) desenvolveram um analisador GCC que utiliza redes neurais, onde cada etapa do analisador é tratada como um problema de classificação multi-classe. Além disso, os autores propuseram um algoritmo iterativo que seleciona automaticamente a melhor análise na fase de treinamento. Eles avaliaram essa estratégia em parte do *corpus* LDC2014T12, alcançando 66,1% na métrica Smatch.

4.1.5. Métodos baseados em aprendizado profundo

Modelos de aprendizado profundo aprendem a produzir grafos AMR a partir de *corpora* anotados sem um processo manual de extração de características. A maioria dos trabalhos adota uma estratégia chamada *sequence-to-sequence* (*seq2seq*) com redes *Bidirectional Long Short-Term Memory* (BiLSTM), convertendo um texto de entrada em um grafo AMR.

Peng et al. (2017) adotaram a estratégia *seq2seq* baseada no trabalho de Vinyals et al. (2015) e propuseram um linearizador e categorizador AMR, visando evitar dados esparsos. Essa estratégia obteve 52% na métrica Smatch sobre o *corpus* LDC2015E86.

van Noord & Bos (2017) utilizaram uma abordagem similar ao trabalho de Peng et al. (2017). Além disso, eles criaram um *corpus* adicional a partir dos analisadores JAMR e CAMR para o treinamento do modelo proposto. Essa abordagem obteve 71% na métrica Smatch sobre o *corpus* LDC2016E25.

Lyu & Titov (2018) introduziram um analisador neural que trata os alinhamentos entre as palavras de uma sentença e os nós do grafo como variáveis latentes e desenvolveram um modelo probabilístico que aprende de ma-

neira conjunta conceitos, relações e alinhamentos. O analisador requer cinco diferentes redes LSTM para identificar os conceitos, relações, a raiz do grafo e os alinhamentos. Esse analisador atingiu 73,7% e 74,4% na métrica Smatch nos *corpora* LDC2015E86 e LDC2016E25, respectivamente.

4.1.6. Métodos baseados em transformers

Introduzido por Vaswani et al. (2017), *transformer* é uma arquitetura que evita recorrência, dependendo de mecanismos de atenção para obter relações entre a entrada e a saída da arquitetura.

Cai & Lam (2020) desenvolveram um modelo fim-a-fim que lida com a tarefa de análise AMR como uma série de decisões duplas sobre uma sentença de entrada, construindo o grafo AMR incrementalmente. Os autores também utilizaram *embeddings* contextuais do modelo de língua BERT (Devlin et al., 2019) para codificar as palavras da sentença de entrada, tendo um ganho de 2,9 pontos na métrica Smatch. A abordagem proposta atingiu 75,4% e 80,2% na métrica Smatch nos *corpora* LDC2014T12 e LDC2017T10, respectivamente.

Bevilacqua et al. (2021) trataram o problema de produzir um grafo AMR a partir de um texto como uma tarefa de transdução simétrica. Eles realizaram uma linearização cuidadosa nos grafos AMR e estenderam o modelo pré-treinado BART (Lewis et al., 2020) para produzir um grafo AMR. O método obteve 84,5% e 80,3% na métrica Smatch nos *corpora* LDC2017T10 e LDC2020T02, respectivamente.

4.2. Analisadores para outras línguas

Os trabalhos anteriores focaram na língua inglesa. Há relativamente poucos trabalhos para outras línguas, principalmente devido à falta de grandes *corpora* anotados, mas algumas iniciativas tentaram superar essa lacuna. Assim como foram organizados os trabalhos para a língua inglesa, aqui os trabalhos também estão organizados pelos seus métodos.

4.2.1. Métodos baseados em regras

Vanderwende et al. (2015) produziram um analisador que pode gerar grafos AMR para sentenças em francês, alemão, espanhol e japonês, onde anotações AMR não foram disponibilizadas. O método converte formas lógicas a partir de um analisador semântico (Vanderwende, 2015) em grafos AMR através de um conjunto de

Analisador	Ano	Abordagem	Corpus - Smatch (%)						
			2013N	2014N	2014	2015	2016	2017	2020
Flanigan et al. (2014)	2014	Grafo	58	-	-	-	-	-	-
Werling et al. (2015)	2015	Grafo	62.3	62.2	-	-	-	-	-
Wang et al. (2015b)	2015	Árvore	63	-	-	-	-	-	-
Wang et al. (2015a)	2015	Árvore	70	70	66	-	-	-	-
Artzi et al. (2015)	2015	GCC	-	66.3	-	-	-	-	-
Goodman et al. (2016)	2016	Árvore	70	-	-	-	-	-	-
Zhou et al. (2016)	2016	Transição	71	71	66	-	-	-	-
Misra & Artzi (2016)	2016	GCC	-	66.1	-	-	-	-	-
Peng et al. (2017)	2017	Aprendizado profundo	-	-	-	52	-	-	-
Damonte et al. (2017)	2017	Transição	-	-	-	64	-	-	-
Konstas et al. (2017)	2017	Aprendizado profundo	-	-	-	62.1	-	-	-
Foland & Martin (2017)	2017	Aprendizado profundo	-	-	-	70.7	-	-	-
Wang & Xue (2017)	2017	Árvore	-	-	68	68.1	-	-	-
Ballesteros & Al-Onaizan (2017)	2017	Transição	-	69	64	-	-	-	-
van Noord & Bos (2017)	2017	Aprendizado profundo	-	-	-	68.5	71	-	-
Peng et al. (2018)	2018	Transição	-	-	-	64	-	-	-
Vilares & Gómez-Rodríguez (2018)	2018	Transição	-	-	-	64	-	-	-
Lyu & Titov (2018)	2018	Aprendizado profundo	-	-	-	73.7	74.4	-	-
Guo & Lu (2018)	2018	Transição	-	74	68.3	68.7	-	69.8	-
Zhang et al. (2019)	2019	Transformer	-	-	70.2	-	-	76.3	-
Cai & Lam (2020)	2020	Transformer	-	-	75.4	-	-	80.2	-
Bevilacqua et al. (2021)	2021	Transformer	-	-	-	-	-	84.5	83

Tabela 2: Resultados dos analisadores AMR para o inglês.

regras. No entanto, os autores não avaliaram o método desenvolvido, pois não existiam *corpora* anotados para esse fim.

Anchiêta & Pardo (2018b) desenvolveram um analisador AMR baseado em regras para o português. Os autores propuseram regras genéricas para converter uma árvore de dependência com informações de papéis semânticos em um grafo AMR. Essa estratégia atingiu 53,3% na métrica Smatch no *corpus* anotado para o português (Anchiêta & Pardo, 2018a).

4.2.2. Métodos baseados árvore

Wang et al. (2018) adaptaram o analisador de Wang et al. (2015b) para o chinês e avaliaram o novo analisador em um *corpus* anotado para o chinês (Li et al., 2016), alcançando 58,7% na métrica Smatch.

4.2.3. Métodos baseados em sistemas de transição

Damonte & Cohen (2018) propuseram uma abordagem baseada em projeção de anotação, que envolve projetar a anotação de uma língua fonte em uma língua alvo. Usando o inglês como língua fonte, os autores produziram grafos AMR para o italiano, espanhol, alemão e chinês. No geral, essa estratégia obteve resultados distantes dos obtidos por analisadores desenvolvidos para o inglês, sendo 43% para o italiano, 42% espanhol, 39% alemão e 35% para o chinês, usando a métrica Smatch.

4.2.4. Métodos baseados em aprendizado profundo

Blloshmi et al. (2020) produziram um analisador AMR multilíngue, adotando técnicas de transferência de aprendizado. O analisador foi modelado como uma abordagem *seq2seq*, onde a camada de codificação é uma rede LSTM bidirecional e a camada de decodificação é uma rede LSTM unidirecional. Na camada de codificação, os autores incluíram *embeddings* multilíngue do BERT, a fim de produzir vetores contextualizados. O método gera grafos AMR para o italiano, espanhol, alemão e chinês, superando os resultados obtidos pelo trabalho de Damonte & Cohen (2018).

4.3. Sumário dos trabalhos

Na Tabela 2, é apresentado um resumo dos trabalhos mencionados para o inglês (uma vez que, para a mesma língua, a comparação é mais direta), organizados por ano de publicação. Os *corpora* 2013N e 2014N referem-se a seção de notícias de LDC2013E117 e LDC2014T12, respectivamente. Os *corpora* 2014, 2015, 2016, 2017 e 2020 são aqueles introduzidos na Seção 3. Na tabela, são destacados os melhores resultados obtidos pelos analisadores para cada *corpus*. É importante notar que os métodos baseados em sistema de transição seguidos pela abordagem baseada em árvore obtiveram melhores resultados para *corpora* pequenos, enquanto estratégias baseadas em aprendizado profundo e *transformers* tiveram melhor performance em *corpora* maiores.

Na próxima seção, serão detalhados os analisadores adaptados para o português, apresentando as melhorias propostas.

5. Adaptação de analisadores AMR para o português

O *corpus* de “O Pequeno Príncipe” anotado para o português (Anchiêta & Pardo, 2018a) tem 1.527 sentenças (alinhadas com a versão em inglês). Dessa forma, adaptaram-se abordagens baseadas em sistemas de transição e árvore, uma vez que essas estratégias atingiram bons resultados em *corpora* pequenos. Apesar disso, adaptou-se também um método baseado em aprendizado profundo, visando aferir seus resultados para o português. Por fim, implementaram-se melhorias no analisador de Anchiêta & Pardo (2018b) desenvolvido para o português.

Para as estratégias baseadas em sistemas de transição e árvore, adaptaram-se os analisadores de Damonte et al. (2017) (AMREager) e Wang et al. (2015b,a) (CAMR), respectivamente, pois eles são *open source* e necessitam apenas de pequenas modificações para reutilização para outra língua. Eles requerem um tokenizador, lematizador, etiquetador morfossintático (*tagger*), analisador de constituintes (*shallow parser*), analisador de dependência (*dependency parser*) e anotador de papéis semânticos. Para o português, essas ferramentas foram providas pelo Stanza (Qi et al., 2020), LX-Parser (Silva et al., 2010) e spaCy⁵. Além disso, os analisadores requerem alguns recursos léxicos, como: lista de países, estados e cidades, palavras negativas e *embeddings* pre-treinadas. Para o português, utilizaram-se as respectivas traduções das listas de palavras e *embeddings* pre-treinadas para o português (Hartmann et al., 2017).

Para treinar esses analisadores, é necessário alinhar os nós do grafo AMR com as palavras da respectiva sentença. Esses analisadores usam o alinhador JAMR (Flanigan et al., 2014) que produz alinhamentos como os mostrados na Figura 4.

```
:: alignments 0-1|0.0 1-2|0 2-3|0.1 3-4|0.3 5-6|0.2
(h / have-03
 :ARG0 (f / flower)
 :ARG1 (t / thorn)
 :purpose (s / spite)
 :mod (j / just))
```

Figura 4: Exemplo de alinhamento entre um grafo AMR e as palavras “*Flowers have thorns just for spite!*”.

O formato do alinhamento é uma lista de *spans* separados por espaço com seu fragmento no grafo, onde cada nó é especificado por um descritor: 0 para o nó raiz, 0.0 para o primeiro filho do nó raiz, 0.1 para o segundo filho do nó raiz e assim por diante. No exemplo da figura, o *span* 0-1 (que é a palavra *Flowers*) está alinhado com o nó 0.0. O JAMR foi desenvolvido para a língua inglesa, não possuindo bom desempenho para o português (Anchiêta & Pardo, 2020a). Por isso, uma primeira adaptação foi utilizar o alinhador desenvolvido por Anchiêta & Pardo (2020a) que foca na língua portuguesa. Apesar disso, esse alinhador não alinha relações de reentrada, ou seja, quando um nó participa de várias relações semânticas no grafo (assim como o JAMR). Para resolver essa questão, os nós que possuem relação de reentrada foram duplicados, como mostrado na Figura 5.

```
:: alignments 1-2|0 0-1|0.0
(s / say-01
 :ARG0 (h / he)
 :ARG1 h)
⇒
:: alignments 1-2|0 0-1|0.0 3-4|0.1
(s / say-01
 :ARG0 (h / he)
 :ARG1 (h1 / he))
```

Figura 5: Duplicação do nó *he* (lado direito) na relação de reentrada para a sentença “*He says to himself.*”.

Além dessa melhoria no alinhamento, propôs-se um aprimoramento no modelo do analisador CAMR. Esse analisador adota um algoritmo baseado no perceptron para aprender características do conjunto de treinamento, usando a métrica Smatch para avaliar e escolher o melhor grafo AMR no conjunto de desenvolvimento. No entanto, como apontado por Song & Gildea (2019) e Anchiêta et al. (2019), a Smatch negligencia vários problemas de análise AMR, atribuindo pontuações altas para sentenças com diferentes significados. Por exemplo, as sentenças “*I have two houses*” e “*She bought three cars*” podem ser representadas como exibido na Figura 6. Usando a métrica Smatch para comparar esses dois grafos, ela retorna 0,29 de medida-f, ou seja, uma pontuação alta para sentenças com significados tão diferentes. Desse modo, o analisador CAMR pode escolher um grafo AMR pior devido à sobrevalorização da métrica Smatch. Apesar da métrica Smatch atribuir pontuações altas para sentenças com significados diferentes, os grafos AMR dessas sentenças compartilham informações em comum. Por exemplo, os grafos da Figura 6 compartilham as mesmas relações: ARG0, :ARG1 e :quant, apesar dos conceitos (nós) entre as relações serem diferentes.

Com o objetivo de ajudar o analisador a escolher um grafo AMR melhor, substitui-se a métrica Smatch pela métrica SEMA (Anchiêta et al., 2019). SEMA é uma métrica mais rigorosa

⁵<https://spacy.io/>

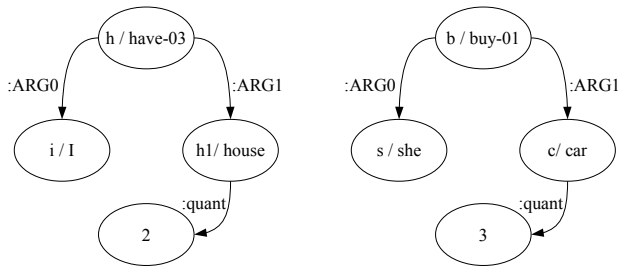


Figura 6: À esquerda, grafo AMR para a sentença “*I have two houses*”; à direita, grafo AMR para a sentença “*She bought three cars.*”

do que a Smatch, fazendo o analisador CAMR atualizar seus parâmetros e buscar um melhor grafo para uma sentença. Para o exemplo da Figura 6, SEMA retorna 0.00 de medida-f. A métrica SEMA compara dois grafos AMR em dois passos. No primeiro, SEMA tenta combinar apenas o nó raiz do grafo de hipótese com o nó raiz do grafo de referência. No segundo passo, a métrica combina o restante dos nós do grafo de hipótese com o restante dos nós do grafo de referência observando as arestas entre os nós, ou seja, se no grafo de hipótese e no grafo de referência houver arestas com o mesmo rótulo, o nó é computado como correto. No artigo de Anchiêta et al. (2019), mostrou-se que essa estratégia torna a SEMA mais rigorosa e mais rápida do que a Smatch.

As adaptações no alinhamento foram aplicadas nos analisadores CAMR e AMREager, enquanto que a SEMA foi aplicada apenas no CAMR.

Além dos analisadores acima, adaptou-se o analisador de van Noord & Bos (2017) (NeuralAMR), utilizando a ferramenta OpenNMT (Klein et al., 2017). Esse analisador adota uma abordagem *seq2seq* com uma rede bidirecional e um mecanismo de atenção geral (Luong et al., 2015). A Tabela 3 mostra os parâmetros do modelo.

Parâmetro	Valor	Parâmetro	Valor
Layers	2	RNN type	brnn
Nodes	500	Dropout	0.3
Epochs	20–25	Vocabulary	100–200
Optimizer	sgd	Max length	750
Learning rate	0.1	Beam size	5
Decay	0.7	Replace unk	true

Tabela 3: Parâmetros do modelo *seq2seq*.

Além desses parâmetros, o modelo aprende *embeddings* durante a fase de treinamento, ou seja, o modelo não usa *embeddings* pre-treinadas porque o *corpus* para o inglês é grande. Mais do que isso, os autores introduziram uma abordagem chamada “*super characters*”, que é a combinação

de palavras e caracteres na camada de entrada. Por exemplo, os autores transformaram relações AMR, como `:ARG0`, em atômicas ao invés de um conjunto de caracteres. Além disso, eles incorporaram classes morfossintáticas nas sentenças de entrada. A Figura 7 apresenta um exemplo dessa estratégia. Além disso, o grafo AMR é linearizado, sendo que as variáveis de cada conceito são removidas, pois o modelo não precisa aprender essas informações. As variáveis são recuperadas em uma etapa de pós-processamento. Por fim, esse modelo não precisa de um alinhador entre as palavras da sentença e um grafo AMR.

Para adaptar esse modelo, como o *corpus* de “O Pequeno Príncipe” é muito menor do que o usado pelos autores do NeuralAMR, adicionaram-se mais informações na estrutura “*super character*”, tais como lema, relações de dependência e entidades nomeadas, com o objetivo de melhorar o analisador. Ademais, utilizaram-se *embeddings* pré-treinadas do português (Hartmann et al., 2017).

Mais do que a adaptação dos analisadores acima, melhorou-se um analisador AMR desenvolvido para o português (Anchiêta & Pardo, 2018b) (RBAMR). Esse analisador lida com o problema de gerar uma estrutura AMR aplicando um conjunto de regras genéricas sobre uma sentença de entrada pré-processada. Os autores desenvolveram seis regras para produzir as seguintes relações AMR: `named entity`, `:mod`, `:manner`, `:degree`, `:polarity` e `:time`. Embora essas relações sejam as mais frequentes no *corpus* de “O Pequeno Príncipe”, os autores não trataram dois fenômenos essenciais que também aparecem com alta frequência: o verbo *ser/estar* e as conjunções. Dessa maneira, estendeu-se o analisador do português para lidar com esses fenômenos, adicionando-se as regras abaixo.

- **Verbo *ser/estar*.** Esta regra cria uma relação `:domain` em sentenças como ... *ser/estar* <SUB> e ... *ser/estar* <ADJ> quando o substantivo ou o adjetivo não for um *frameset* do PropBank. A Figura 8 mostra um exemplo dessa regra.
- **Conjunções.** Esta regra lida com dois tipos de conjunções: contrastiva e aditiva. Na primeira, produz-se o conceito `contrast-01` para as seguintes conjunções: *mas*, *enquanto*, *enquanto que*, *no entanto*, *entretanto* e assim por diante. Na segunda, criam-se os conceitos `and` ou `or`, como apresentado na Figura 9.

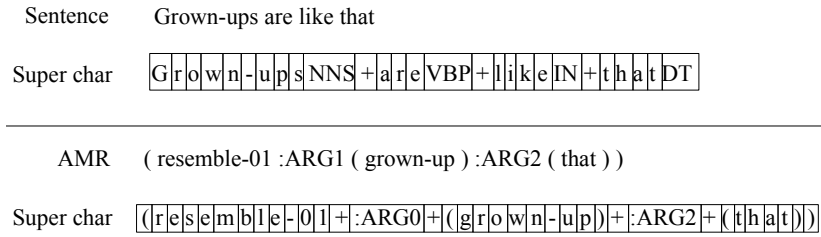


Figura 7: Sentença de entrada “*Grown-ups are like that*” com etiquetas e grafo AMR (`resemble-01 :ARG1 (grown-up) :ARG2 (that)`). O símbolo + representa espaço.

Além dessas duas regras, desenvolveu-se um método de poda para os grafos AMR em uma etapa de pós-processamento com o objetivo de aumentar a qualidade dos grafos. O analisador RBAMR não mantém os traços de quais nós já foram produzidos, gerando redundância. Portanto, o método de poda remove todos os nós duplicados que possuem o mesmo nó pai.

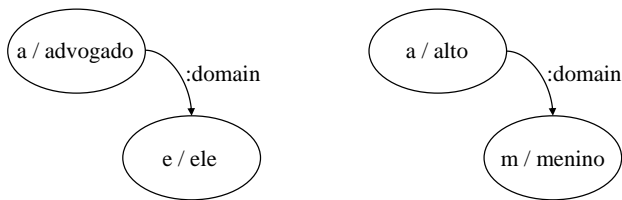


Figura 8: Regra para o verbo ser/estar para as sentenças: “*Ele é um advogado.*” (esquerda) e “*O menino é alto.*” (direita).

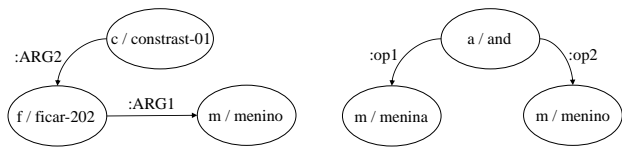


Figura 9: Regras para conjunções: conceitos `contrast-01` e `and`, respectivamente, para as sentenças: “*Mas o menino ficou.*” (esquerda) e “*A menina e o menino.*” (direita).

Como foi realizada uma duplicação nos nós com reentrada gerados pelos analisadores CAMR e AMREager, no pós-processamento recuperou-se a relação de entrada. A Figura 10 exemplifica esse processo.

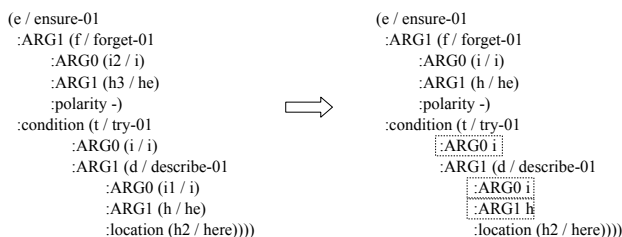


Figura 10: Um exemplo de nós com reentrada recuperados. Na esquerda, um exemplo com nós duplicados; na direita, os nós recuperados `i` e `he`.

Para o analisador RBAMR, aplicou-se um método de poda para remover nós redundantes que possuem o mesmo nó pai. Na Figura 11, é exibido um exemplo do método.

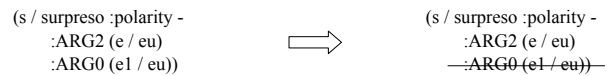


Figura 11: Na esquerda, nó redundante; na direita, nó podado `eu`.

No que segue, relata-se a avaliação dos analisadores, bem com uma detalhada análise de erros sobre um *corpus* anotado para o português.

6. Avaliação e resultados

Com o objetivo de avaliar os analisadores adaptados, conduziu-se um experimento sobre o *corpus* anotado de “O Pequeno Príncipe” (Anchiêta & Pardo, 2018a). O *corpus* está alinhado com a versão em inglês do livro, mantendo as divisões de treinamento, desenvolvimento e teste iguais ao do inglês⁶, com 1.274, 145 e 143 sentenças, respectivamente. Além disso, separaram-se as sentenças do conjunto de teste pelo tamanho, pois, quanto maior a sentença, mais desafiadora é a análise, pois os erros produzidos em etapas de pré-processamento são propagados para a geração dos grafos. Portanto, calculou-se a média do tamanho das sentenças, obtendo um valor de 10,46 palavras por sentença, sendo 80 sentenças maiores que a média e 63 sentenças menores que a média. A separação por tamanho visou explicitar os desafios da análise.

Para avaliar os analisadores, as métricas Smatch e SEMA foram utilizadas. Nas Figuras 12 e 13, são apresentados os resultados das métricas para sentenças curtas e longas, respectivamente.

Nessas figuras, pode-se ver que o analisador RBAMR obteve os melhores resultados em ambas as métricas e tamanhos de sentença. O analisador atingiu 0,66 e 0,48 de medida-f para sen-

⁶<https://amr.isi.edu/download.html>

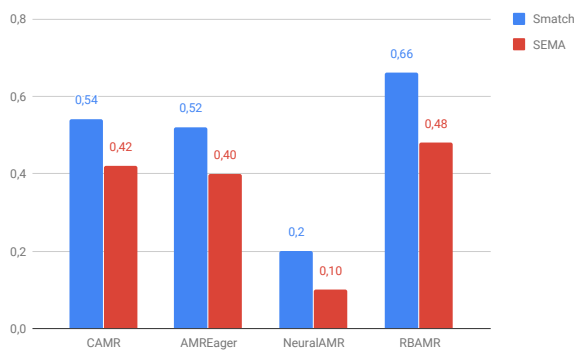


Figura 12: Resultados para sentenças curtas.

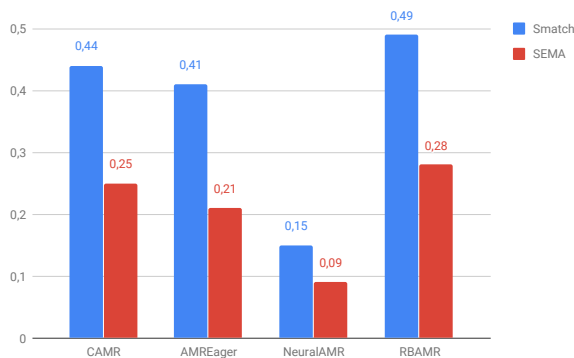


Figura 13: Resultados para sentenças longas.

tenças curtas na métrica Smatch e SEMA, respectivamente, e 0,49 e 0,28 para sentenças longas nas mesmas métricas. Os analisadores CAMR e AMREager tiveram um desempenho similar em ambas as métricas e tamanhos de sentenças, com o primeiro sendo levemente superior ao segundo. Por fim, o analisador NeuralAMR não alcançou bons resultados. Acredita-se que seja porque o *corpus* é pequeno.

A Tabela 4 apresenta uma análise detalhada dos resultados, mostrando as contribuições das adaptações propostas para cada analisador, exibindo valores de medida-f da Smatch e da SEMA. A partir dessa tabela, pode-se ver que as novas regras mais o método de poda melhoraram a versão anterior do RBAMR em 0,05 e 0,03 na métrica Smatch para sentenças curtas e longas, respectivamente. A melhoria no alinhamento também superou a versão original dos analisadores CAMR e AMREager. Além disso, adotar a métrica SEMA na fase de treinamento do CAMR produziu melhores resultados. É interessante perceber que modificações relativamente simples podem gerar resultados melhores.

Observando-se os grafos gerados pelos analisadores CAMR e AMREager, notou-se que um dos fatores para o resultado ser pior do que o RBAMR são os erros no alinhamento. Embora tenha sido

utilizado um alinhador desenvolvido para o português, alguns fenômenos são difíceis de se lidar, como o sujeito oculto. Por exemplo, na sentença “Preciso é de um carneiro.”, o sujeito ‘eu’ não aparece na sentença, mas ele foi anotado, conforme exibido na Figura 14. Consequentemente, o nó *eu* não possui um correspondente na sentença.

```
:: alignments 0-1|0 4-5|0.1
Preciso é de um carneiro
(p / precisar-01
 :ARG0 (e / eu)
 :ARG1 (c / carneiro))
```

Figura 14: Alinhamento entre um grafo AMR e palavras para a sentença “Preciso é de um carneiro.”.

Além da análise nos alinhamentos, observou-se também que o analisador CAMR produz relações *:null_edge* sempre que ele não identifica uma relação adequada. Esse caso ocorreu 95 vezes, representando 20% do número total de relações. Se o analisador trocasse a relação *:null_edge* para a relação *:ARG0* (já que ela é bastante frequente), por exemplo, o resultado do analisador melhoraria 5% na medida-f. Ademais, o analisador também produz conceitos *null_tag* quando ele não identifica um conceito na sentença. Esses casos apareceram apenas duas vezes nos resultados, e são mais difíceis de se tratar. Na Figura 15, é mostrado um exemplo desses problemas. Acredita-se que essas questões surgiram devido ao *corpus* ser pequeno, uma vez que esses problemas não aparecem em *corpora* maiores do inglês.

```
(d / deitado
 :null_edge (n / null_tag
 :time (d1 / dia
 :mod (m / meio)))
 :ARG1 (s / sol
 :mod (t / todo))
 :ARG1 (m1 / mundo)
 :null_edge (f / frança))
```

Figura 15: Relação *:null_edge* e conceito *null_tag* produzidos pelo CAMR para a sentença “Quando é meio dia nos Estados Unidos, o sol, todo mundo sabe, está se deitando na França.”.

O analisador AMREager gera conceitos *emptygraph* quando ele não identifica conceitos e relações em uma sentença. Esse problema ocorreu seis vezes nos resultados e foi um dos responsáveis pelo baixo resultado na identificação de conceitos e relações de papéis semânticos quando comparado ao analisador CAMR, conforme apresentado na Figura 16. Por exemplo, para a sentença “Um dia eu vi o sol se pôr quarenta e três vezes!”, o AMREager gera

Analisador	Sentenças curtas		Sentenças longas	
	Smatch	SEMA	Smatch	SEMA
RBAMR	0,61	0,43	0,46	0,25
RBAMR + regras	0,62	0,44	0,48	0,27
RBAMR + poda	0,64	0,46	0,47	0,26
RBAMR + regras + poda	0,66	0,48	0,49	0,28
CAMR	0,51	0,40	0,40	0,22
CAMR + alinhamento	0,53	0,41	0,42	0,24
CAMR + SEMA	0,52	0,41	0,41	0,23
CAMR + alinhamento + SEMA	0,54	0,42	0,44	0,25
AMREager	0,50	0,39	0,39	0,19
AMREager + alinhamento	0,52	0,40	0,41	0,21

Tabela 4: Contribuições das adaptações.

apenas `emptygraph`. Acredita-se também que esse problema seja devido ao tamanho do *corpus*.

O analisador `NeuralAMR`, embora não produza relações `:null_edge` e nem conceitos `null_tag` e `emptygraph`, não teve boa performance. Acredita-se também que esse problema seja devido ao tamanho do *corpus*, pois modelos *seq2seq* requerem um vocabulário grande para atingirem resultados satisfatórios.

O analisador `RBAMR` obteve os melhores resultados. Esse analisador é baseado em regras e não requer alinhamento entre as palavras de uma sentença e os nós do grafo. Além disso, ele usa apenas duas ferramentas para obter informações morfossintáticas e papéis semânticos, evitando muitos erros na fase de pré-processamento.

Além dos erros destacados acima, analisaram-se erros na geração de nós (conceitos) e arestas (relações) a fim de obter mais *insights*. Como um grafo AMR possui várias relações possíveis (há mais de 100 previstas no formalismo AMR), analisar cada relação é uma tarefa laboriosa. Portanto, dividiu-se a tarefa de análise em duas sub-tarefas para facilitar a análise: identificação de conceitos e papéis semânticos. Para isso, usou-se a ferramenta de avaliação de Damonte et al. (2017) para visualizar a performance desses componentes.

Na Figura 16, é exibido o resultado para esses componentes. A partir dessa figura, pode-se ver que o `RBAMR` atingiu os melhores resultados em ambos os componentes. Apesar dos analisadores adaptados não terem alcançado resultados melhores do que o `RBAMR` no *corpus* analisado, acredita-se que em um *corpus* maior eles possam ter resultados superiores.

Por fim, realizou-se uma análise de etiquetas morfossintáticas identificadas automaticamente nas 143 sentenças do conjunto de teste. Nessa

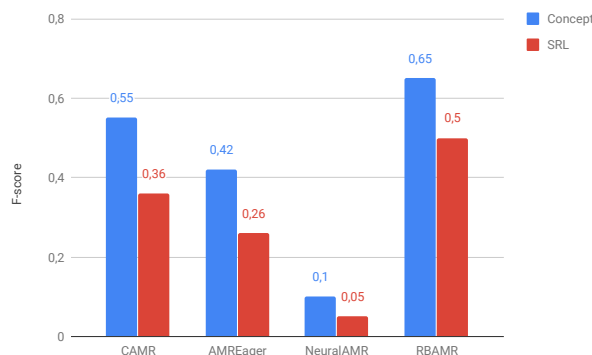


Figura 16: Resultados para identificação de conceitos e papéis semânticos.

análise, observou-se que o *parser* errou 90 etiquetas morfossintáticas, sendo a maioria em sentenças com mais de dez palavras. Isso justifica o baixo resultado em sentenças com mais de dez palavras. A etiqueta que o *parser* mais errou foi o verbo.

7. Conclusão

Neste artigo, apresentaram-se adaptações com melhorias de alguns analisadores AMR do inglês para o português. Além disso, melhorou-se o analisador AMR desenvolvido para o português através do desenvolvimento de novas regras. Essas ferramentas foram avaliadas usando o *corpus* de “O Pequeno Príncipe” anotado na língua portuguesa. Mais do que isso, apresentou-se uma análise detalhada dos resultados, mostrando as contribuições das adaptações realizadas e uma detalhada análise de erros a fim de prover *insights* para trabalhos futuros.

Embora soluções baseadas em aprendizado de máquina e, mais recentemente, abordagens baseadas em *transformers* estejam atingindo resulta-

dos impressionantes, elas requerem grandes *corpora* anotados. Para línguas com poucos recursos semânticos, como o português, grandes *corpora* ainda não são uma realidade. Nesses casos, abordagens baseadas em regras ainda têm valor significativo. Ademais, métodos adaptados podem atingir resultados interessantes, ajudando a produzir os primeiros recursos e ferramentas para uma língua.

Como trabalho futuro, pretende-se aumentar o *corpus*, adotando uma estratégia baseada em *ensemble* e *back-translation* com base nos analisadores desenvolvidos e desenvolver um método para tornar explícito o sujeito oculto.

Mais informações sobre esse trabalho e os recursos e ferramentas desenvolvidos podem ser encontrados nos portais web dos projetos OPINANDO⁷ e POeTiSA⁸.

Agradecimentos

Os autores agradecem ao Instituto Federal do Piauí e ao Centro de Inteligência Artificial (C4AI - <http://c4ai.inova.usp.br/>) da Universidade de São Paulo pelo apoio a este trabalho.

Referências

- Abend, Omri & Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). Em *51st Annual Meeting of the Association for Computational Linguistics*, 228–238.
- Anchiêta, Rafael & Thiago Pardo. 2018a. Towards AMR-BR: A SemBank for Brazilian Portuguese language. Em *11th International Conference on Language Resources and Evaluation (LREC)*, 974–979.
- Anchiêta, Rafael & Thiago Pardo. 2020a. Semantically inspired AMR alignment for the Portuguese language. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1595–1600. [doi 10.18653/v1/2020.emnlp-main.123](https://doi.org/10.18653/v1/2020.emnlp-main.123).
- Anchiêta, Rafael T., Marco Antonio Sobrevilla Cabezudo & Thiago A. S. Pardo. 2019. SEMA: an extended semantic evaluation metric for AMR. *CoRR* abs/1905.12069. arXiv.
- Anchiêta, Rafael Torres & Thiago Alexandre Salgueiro Pardo. 2018b. A rule-based AMR parser for Portuguese. Em *16th Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, 341–353. [doi 10.1007/978-3-030-03928-8_28](https://doi.org/10.1007/978-3-030-03928-8_28).
- Anchiêta, Rafael Torres & Thiago Alexandre Salgueiro Pardo. 2020b. Exploring the potentiality of semantic features for paraphrase detection. Em *14th International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 228–238. [doi 10.1007/978-3-030-41505-1_22](https://doi.org/10.1007/978-3-030-41505-1_22).
- Artzi, Yoav, Kenton Lee & Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1699–1710. [doi 10.18653/v1/D15-1198](https://doi.org/10.18653/v1/D15-1198).
- Ballesteros, Miguel & Yaser Al-Onaizan. 2017. AMR parsing using stack-LSTMs. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1269–1275. [doi 10.18653/v1/D17-1130](https://doi.org/10.18653/v1/D17-1130).
- Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer & Nathan Schneider. 2013. Abstract meaning representation for Sembanking. Em *7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186.
- Bender, Emily M. & Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. Em *58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. [doi 10.18653/v1/2020.acl-main.463](https://doi.org/10.18653/v1/2020.acl-main.463).
- Bevilacqua, Michele, Rexhina Blloshmi & Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. Em *35th AAAI Conference on Artificial Intelligence*, 12564–12573.
- Blloshmi, Rexhina, Rocco Tripodi & Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2487–2500. [doi 10.18653/v1/2020.emnlp-main.195](https://doi.org/10.18653/v1/2020.emnlp-main.195).
- Burns, Gully A., Ulf Hermjakob & José Luis Ambite. 2016. Abstract meaning representations as linked data. Em *15th International Semantic Web Conference (ISWC)*, 12–20.
- Cai, Deng & Wai Lam. 2020. AMR parsing via graph-sequence iterative inference.

⁷<https://sites.google.com/icmc.usp.br/opinando/>

⁸<https://sites.google.com/icmc.usp.br/poetisa>

- Em *58th Annual Meeting of the Association for Computational Linguistics*, 1290–1301. doi 10.18653/v1/2020.acl-main.119.
- Cai, Shu & Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. Em *51st Annual Meeting of the Association for Computational Linguistics*, 748–752.
- Damonte, Marco & Shay B. Cohen. 2018. Cross-lingual abstract meaning representation parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1146–1155. doi 10.18653/v1/N18-1104.
- Damonte, Marco, Shay B. Cohen & Giorgio Satta. 2017. An incremental parser for abstract meaning representation. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 536–546.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 4171–4186. doi 10.18653/v1/N19-1423.
- Flanigan, Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer & Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. Em *52nd Annual Meeting of the Association for Computational Linguistics*, 1426–1436. doi 10.3115/v1/P14-1134.
- Foland, William & James H. Martin. 2017. Abstract meaning representation parsing using LSTM recurrent neural networks. Em *55th Annual Meeting of the Association for Computational Linguistics*, 463–472. doi 10.18653/v1/P17-1043.
- Goodman, James, Andreas Vlachos & Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. Em *54th Annual Meeting of the Association for Computational Linguistics*, 1–11. doi 10.18653/v1/P16-1001.
- Guo, Zhijiang & Wei Lu. 2018. Better transition-based AMR parsing with a refined search space. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1712–1722. doi 10.18653/v1/D18-1198.
- Hardy, Hardy & Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using abstract meaning representation. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 768–773. doi 10.18653/v1/D18-1086.
- Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Silva & Sandra Aluísio. 2017. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. Em *11th Brazilian Symposium in Information and Human Language Technology*, 122–131.
- Issa, Fuad, Marco Damonte, Shay B. Cohen, Xiaohui Yan & Yi Chang. 2018. Abstract meaning representation for paraphrase detection. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 442–452. doi 10.18653/v1/N18-1041.
- Jurafsky, Dan & James H. Martin. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. Prentice Hall.
- Kingsbury, Paul & Martha Palmer. 2002. From TreeBank to PropBank. Em *3rd International Conference on Language Resources and Evaluation (LREC)*, 1989–1993.
- Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart & Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. Em *Proceedings of ACL 2017, System Demonstrations*, 67–72.
- Konstas, Ioannis, Srinivasan Iyer, Mark Yatskar, Yejin Choi & Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. Em *55th Annual Meeting of the Association for Computational Linguistics*, 146–157. doi 10.18653/v1/P17-1014.
- Lehmann, Fritz. 1992. *Semantic networks in artificial intelligence*. Elsevier Science Inc.
- Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov & Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Em *58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. doi 10.18653/v1/2020.acl-main.703.
- Li, Bin, Yuan Wen, Weiguang Qu, Lijun Bu & Nianwen Xue. 2016. Annotating

- the little prince with Chinese AMRs. Em *10th Linguistic Annotation Workshop*, 7–15. doi 10.18653/v1/W16-1702.
- Liu, Fei, Jeffrey Flanigan, Sam Thomson, Norman Sadeh & Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1077–1086. doi 10.3115/v1/N15-1114.
- Luong, Thang, Hieu Pham & Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1412–1421. doi 10.18653/v1/D15-1166.
- Lyu, Chunchuan & Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. Em *56th Annual Meeting of the Association for Computational Linguistics*, 397–407. doi 10.18653/v1/P18-1037.
- Matthiessen, Christian & John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from english and japanese*. Pinter Publishers.
- Migueles-Abraira, Noelia, Rodrigo Agerri & Arantza Diaz de Ilarraza. 2018. Annotating abstract meaning representations for Spanish. Em *11th International Conference on Language Resources and Evaluation (LREC)*, 3074–3078.
- Misra, Dipendra Kumar & Yoav Artzi. 2016. Neural shift-reduce CCG semantic parsing. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1775–1786. doi 10.18653/v1/D16-1183.
- Mitra, Arindam & Chitta Baral. 2016. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. Em *AAAI Conference on Artificial Intelligence*, 2779–2785. doi 10.1609/aaai.v30i1.10354.
- Nivre, Joakim. 2004. Incrementality in deterministic dependency parsing. Em *Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, 50–57.
- van Noord, Rik & Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal* 7. 93–108.
- Osa, Takayuki, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel & Jan Peters. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics* 7(1-2). 1–179. doi 10.1561/23000000053.
- Palmer, Martha, Daniel Gildea & Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1). 71–106. doi 10.1162/0891201053630264.
- Pan, Xiaoman, Taylor Cassidy, Ulf Hermjakob, Heng Ji & Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1130–1139. doi 10.3115/v1/N15-1119.
- Peng, Xiaochang, Daniel Gildea & Giorgio Satta. 2018. AMR parsing with cache transition systems. Em *32nd AAAI Conference on Artificial Intelligence*, 4897–4904. doi 10.1609/aaai.v32i1.11922.
- Peng, Xiaochang, Chuan Wang, Daniel Gildea & Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 366–375.
- Pereira, Fernando CN & Stuart M Shieber. 2002. *Prolog and natural-language analysis*. Microtome Publishing.
- Pourdanghani, Nima, Kevin Knight & Ulf Hermjakob. 2016. Generating English from abstract meaning representations. Em *9th International Natural Language Generation Conference*, 21–25. doi 10.18653/v1/W16-6603.
- Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton & Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. Em *58th Annual Meeting of the Association for Computational Linguistics*, 101–108.
- Silva, João, António Branco, Sérgio Castro & Ruben Reis. 2010. Out-of-the-box robust parsing of portuguese. Em *9th International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 75–85. doi 10.1007/978-3-642-12320-7_10.
- Sobrevilla Cabezudo, Marco Antonio & Thiago Pardo. 2019. Towards a general abstract meaning representation corpus for Brazilian Portuguese. Em *13th Linguistic Annotation Workshop*, 236–244. doi 10.18653/v1/W19-4028.

- Song, Linfeng & Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. Em *57th Annual Meeting of the Association for Computational Linguistics*, 4547–4552. doi 10.18653/v1/P19-1446.
- Song, Linfeng, Daniel Gildea, Yue Zhang, Zhiguo Wang & Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics* 7. 19–31. doi 10.1162/tacl_a_00252.
- Song, Linfeng, Xiaochang Peng, Yue Zhang, Zhiguo Wang & Daniel Gildea. 2017. AMR-to-text generation with synchronous node replacement grammar. Em *55th Annual Meeting of the Association for Computational Linguistics*, 7–13. doi 10.18653/v1/P17-2002.
- Song, Linfeng, Yue Zhang, Zhiguo Wang & Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. Em *56th Annual Meeting of the Association for Computational Linguistics*, 1616–1626. doi 10.18653/v1/P18-1150.
- Steedman, Mark. 1996. *Surface structure and interpretation*. MIT Press.
- Steedman, Mark. 2001. *The syntactic process*. MIT Press.
- Uchida, Hiroshi, Meiyong Zhu & Tarcisio Della Senta. 2006. *UNL: Universal networking language*. UNDL Foundation, International Environment House.
- Vanderwende, Lucy. 2015. NLPwin—an introduction. Relatório técnico. Microsoft Research tech report no. MSR-TR-2015-23.
- Vanderwende, Lucy, Arul Menezes & Chris Quirk. 2015. An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus. Em *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 26–30. doi 10.3115/v1/N15-3006.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. Em *31st International Conference on Neural Information Processing Systems*, 5998–6008.
- Vilares, David & Carlos Gómez-Rodríguez. 2018. A transition-based algorithm for unrestricted AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 142–149. doi 10.18653/v1/N18-2023.
- Vinyals, Oriol, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever & Geoffrey Hinton. 2015. Grammar as a foreign language. Em *Advances in neural information processing systems*, 2773–2781.
- Wang, Chuan, Bin Li & Nianwen Xue. 2018. Transition-based Chinese AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 247–252. doi 10.18653/v1/N18-2040.
- Wang, Chuan & Nianwen Xue. 2017. Getting the most out of AMR parsing. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1257–1268. doi 10.18653/v1/D17-1129.
- Wang, Chuan, Nianwen Xue & Sameer Pradhan. 2015a. Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. Em *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 857–862. doi 10.3115/v1/P15-2141.
- Wang, Chuan, Nianwen Xue & Sameer Pradhan. 2015b. A transition-based algorithm for AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 366–375. doi 10.3115/v1/N15-1040.
- Werling, Keenon, Gabor Angeli & Christopher D. Manning. 2015. Robust subgraph generation improves abstract meaning representation parsing. Em *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 982–991. doi 10.3115/v1/P15-1095.
- Zhang, Sheng, Xutai Ma, Kevin Duh & Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. Em *57th Annual Meeting of the Association for Computational Linguistics*, 80–94. doi 10.18653/v1/P19-1009.
- Zhou, Junsheng, Feiyu Xu, Hans Uszkoreit, Weiguang Qu, Ran Li & Yanhui Gu. 2016. AMR parsing with an incremental joint model. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 680–689. doi 10.18653/v1/D16-1065.