UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Abstract Meaning Representation Parsing for the Brazilian Portuguese Language

Rafael Torres Anchiêta

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Rafael Torres Anchiêta

Abstract Meaning Representation Parsing for the Brazilian Portuguese Language

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Thiago Alexandre Salgueiro Pardo

USP – São Carlos
July 2020

Rafael Torres Anchiêta

Analisadores para Representação Abstrata de Significado para o Português Brasileiro

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Thiago Alexandre Salgueiro Pardo

USP – São Carlos
Julho de 2020

ACKNOWLEDGEMENTS

First of all, I would like to thank God because He loved me first. I thank Him for the grace given to me. I thank Him for a loving wife, kind parents, and an excellent advisor.

I would like to thank my wife, Alissa because she is loving and caring. I thank her for reading all my papers, for listening to all my presentations, ..., for her unconditional support.

I would like to thank my parents, Adalberto and Maria Vandeci, honest people that ever supported me in the studies. I thank my brothers, Virna and Cahuê, for their teachings.

I would like to thank deeply my advisor, Thiago Pardo, for has provided valuable pieces of advice and guidance and for all its teachings.

I would like to thank my friends from 2º IPB de Uberaba, IPB de São Carlos, ICE em Bequimão, and Teresina for praying for me.

Of course, I would like to thank the University of São Paulo, Institute of Mathematics and Computer Sciences, Interinstitutional Center for Computational Linguistics, and Federal Institute of Piauí for financial support and structural.

RESUMO

ANCHIÊTA, R. T. **Analisadores para Representação Abstrata de Significado para o Português Brasileiro**. 2020. 142 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Semântica computacional é a área encarregada de estudar possíveis representações semânticas, ou seja, formalismos semânticos que são computacionalmente viáveis para representar expressões da língua humana. Esses formalismos desempenham um papel importante para o entendimento de uma língua natural, capturando o significado de expressões linguísticas. Além disso, eles são o principal ingrediente para desenvolver analisadores semânticos, que são responsáveis por mapear sentenças de uma língua natural em uma representação semântica computacionalmente tratável. Com o objetivo de representar e entender características semânticas de uma língua natural e, com isso, desenvolver ferramentas computacionais que produzam resultados mais próximos aos dos humanos, diversos formalismos semânticos foram propostos, como: Universal Networking Language (UNL), Universal Conceptual Cognitive Annotation, (UCCA), Abstract Meaning Representation (AMR), entre outros. Em especial, Abstract Meaning Representation (AMR) é um formalismo semântico baseado em grafo direcionado que possui única raiz com nós e arestas rotulados. Os nós representam conceitos (que podem ser as palavras de uma sentença), as arestas representam relações semânticas entre os conceitos e os nós não possuem alinhamento explícito com as palavras da sentença. AMR compreende algumas caractetísticas semânticas como: entidades nomeadas, correferência, papéis semânticos, desambiguação lexical, entre outras. Neste trabalho, focou-se na representação AMR para a língua portuguesa, pois ela possui uma estrutura mais fácil de produzir do que outras representações semânticas. Dessa forma, anotou-se o livro do Pequeno Príncipe, que é primeiro corpus anotado nesse formalismo para a língua portuguesa e desenvolveu-se o primeiro analisador semântico para essa representação. Além disso, adaptou-se alguns métodos de análise semântica da língua inglesa para a língua portuguesa. Mais do que isso, desenvolveu-se um novo método de alinhamento entre as palavras da sentença e os nós do grafo que melhora os resultados dos analisadores semânticos adaptados e um novo método de avaliação entre grafos AMRs que é mais robusto, rápido e justo do que a métrica tradicional de avaliação. Por fim, utilizou-se esses métodos em uma tarefa de detecção de paráfrase, combinando tanto características semânticas implícitas quanto explícitas para classificar se uma sentença é paráfrase de outra.

Palavras-chave: Representação Abstrata de Significado, Análise Semântica, Anotação Semântica.

ABSTRACT

ANCHIÊTA, R. T. **Abstract Meaning Representation Parsing for the Brazilian Portuguese Language**. 2020. 142 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Computational semantics is the area in charge of studying possible meaning representations, that is, computationally viable semantic formalisms to represent human expressions. Such formalisms play an important role in making sense of natural language, capturing the meaning of linguistic statements. Moreover, these formalisms are the main component to develop semantic parsers, which are responsible to map sentences of a natural language into a computationally treatable meaning representation. In order to represent and understand semantic features of a natural language and, with that, develop computational tools that produce results close to those of humans, several semantic formalisms were proposed, as Universal Networking Language (UNL), Universal Conceptual Cognitive Annotation (UCCA), Abstract Meaning Representation (AMR), among others. In special, AMR is a rooted directed graph-based semantic formalism with labeled nodes and edges. The nodes are concepts (that may be the words of a sentence) and the edges are semantic relations among them, where the nodes do not have an explicit alignment with the tokens of the sentences. Furthermore, AMR encompasses some linguistic features, as named entities, coreference, semantic roles, word sense disambiguation, and others. In this work, we focused on AMR representation for Portuguese, since it has a simpler structure to produce than other semantic formalisms. In this way, we annotated the Little Prince book, which is the first annotated corpus with AMR information for Portuguese and developed the first AMR parser for Portuguese. Moreover, we adapted some AMR parsing methods from English to Portuguese. More than that, we developed a new alignment strategy to align the word tokens of the sentence and the nodes of the AMR graph that improves the results of the adapted AMR parsers and a new metric to evaluate AMR graphs, which is more robust, faster, and fairer than the traditional AMR metric. Finally, we used these resources and methods in a paraphrase detection task, joining both explicit and implicit semantic features to classify if two sentences are paraphrase each other.

Keywords: Abstract Meaning Representation, Semantic Parsing, Semantic Annotation.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER

1

INTRODUCTION

Natural Language Processing (NLP) is a subfield of Artificial Intelligence (AI), which aims to get computers to perform useful tasks involving human language (JURAFSKY; MARTIN, 2009). For that, computers must have some knowledge of language, such as phonetics and phonology, morphology, syntax, semantics, pragmatics, and discourse. This doctoral dissertation focuses on semantics, which comprises knowledge of meaning.

In general, semantics may be divided into two levels: words and sentences. The first is in charge of understanding the lexicon (or units) of a sentence, while, in the second, the meaning of a sentence may be determined by the meaning of its parts and how they are combined (i.e., compositionally) or the understanding of a sentence may be determined directly (i.e., non-compositionally) (SAEED, 2016). In special, sentence-level semantics, which is the focus of this dissertation, has gained the attention of the NLP community, since it may be used in many NLP applications, such as text generation, automatic summarization, machine translation, and others. These tasks need a deeper understanding of the text to produce results more similar way to how humans do (ABEND; RAPPOPORT, 2017; ABZIANIDZE; BOS, 2019).

In particular, computational semantics is the area responsible for studying possible meaning representations (i.e., computationally viable representations) for human language expressions (BLACKBURN; BOS, 2003; JURAFSKY; MARTIN, 2009). A meaning representation is a formal structure designed to capture the meaning of linguistic expressions, abstracting away from the syntactic structure, and intending to be language-independent (JURAFSKY; MARTIN, 2009). Moreover, it is one of the most important components in semantic parsing, which is the task of translating a natural language into a meaning representation.

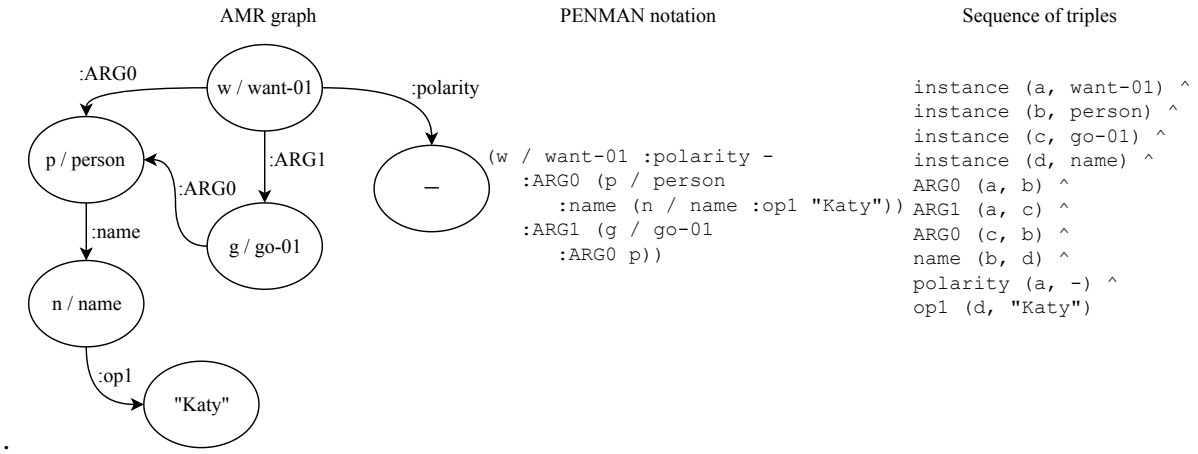
Intending to produce better NLP applications through a deeper understanding of a text, several representations with different structures and levels (shallow to deeper) were developed, as the traditional First-Order Logic (FOL), detailed in Jurafsky and Martin (2009), Semantic Networks (SN) (LEHMANN, 1992), Universal Networking Language (UNL) (UCHIDA; ZHU;

SENTA, 1996), Universal Conceptual Cognitive Annotation (UCCA) (ABEND; RAPPOPORT, 2013), and Abstract Meaning Representation (AMR) (BANARESCU *et al.*, 2013), among others. Here, AMR, which is the language of interest, is briefly introduced.

1.0.1 Fundamentals of AMR

AMR attracted the attention of the NLP community due to its simpler structure compared to other representations (BOS, 2016). It is a graph-based representation where it refers to nodes as concepts, to edges as relations among concepts, and to edge labels as roles. In addition, AMR may be represented as a serialized graph in PENMAN notation (MATTHIESSEN; BATEMAN, 1991) and a sequence of triples, as depicted in Figure 1. In this figure, `want-01` is the root of the graph, the `w` prefix is a variable that may be used in reentrancy relations (multiple incoming edges), and the `01` suffix is the sense of the concept from the PropBank lexicon (KINGSBURY; PALMER, 2002). The `person` node indicates the named entity "`Katy`" and the '`-`' node is a constant value, as it gets no variable. Moreover, `:ARGx` relations are predicates from the PropBank resource, which encode semantic information according to each PropBank sense. The `:ARG0` relation between `go-01` and `person` nodes is a reentrancy relation since the `person` node is re-used in the structure. The `:polarity`, `:name` and `:op1` relations are unique of the AMR language, `:polarity` and `:op1` are characterized as attributes, as their targets are constants: '`-`' and "`Katy`", respectively. Overall, AMR has over 100 relations. The guidelines[1] and AMR original paper (BANARESCU *et al.*, 2013) give more details about other relations.

Figure 1 – An example of AMR notation for the sentence "Katy does not want to go."



Source – Adapted from Banarescu *et al.* (2013)

As one can see from the above example, AMR explicitly characterizes the semantic information through a graph-structure. In Figure 1, the following semantic features are shown: named entity (`person -> name -> Katy`), word sense disambiguation (`-01` suffix), coreference (`:ARG0` relation between `go-01` and `person` nodes), semantic roles (`ARG0`, `ARG1`), and

[1] <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>

negation (`:polarity` relation). This explicit information is the main difference in AMR when compared to more recent vector representations, as Word2Vec (MIKOLOV *et al.*, 2013), GLOVE (PENNINGTON; SOCHER; MANNING, 2014), BERT (DEVLIN *et al.*, 2019), and others. In the latter, the semantic information is implicitly encoded through vectors of real-valued numbers that represent particular character, word or sentence. One advantage of explicit information is to turn the representation more interpretable for humans compared to implicit information. This fostered the growth of applications in the area of Natural Language Understanding (NLU), such as question answering (SACHAN; XING, 2016; MITRA; BARAL, 2016), summarization (LIU *et al.*, 2015; LIAO; LEBANOFF; LIU, 2018; HARDY; VLACHOS, 2018), text generation (SONG *et al.*, 2018; ZHU *et al.*, 2019; RIBEIRO; GARDENT; GUREVYCH, 2019), machine translation (SONG *et al.*, 2019), and others. On the other hand, explicit information is more difficult to deal with than vector representations, since it requires much more annotated data to produce good computational linguistic tools.

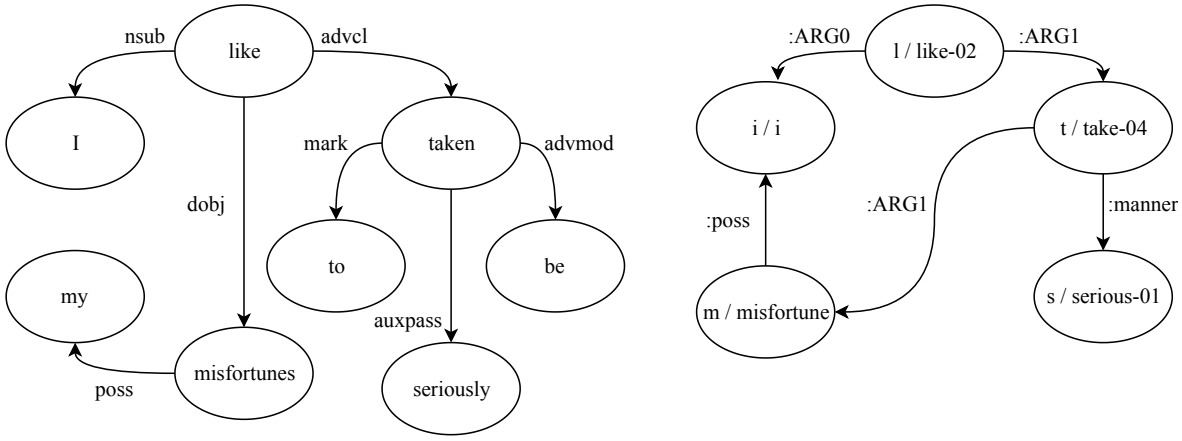
The above mentioned applications require an AMR parser (or analyzer) to represent natural language statements into AMR structures. The objective of a semantic parser is to abstract away from syntactic phenomena, eliminating ambiguous interpretations and focusing on semantic features, as named entities, coreference, word sense disambiguation, semantic roles, among others (GOODMAN; VLACHOS; NARADOWSKY, 2016). Several authors proposed a variety of AMR parsing methods to achieve that aim. These methods are based on rules, dependency-tree, transition-system, and deep learning.

Rules and dependency tree-based methods explore the hypothesis that an AMR structure is relatively similar to the dependency tree structure, since both describe relations between a parent and their children, or between a head and their dependents, as shown in Figure 2. One can see that, generally, content words as `I`, `like`, `taken`, `seriously`, and `misfortunes` become AMR concepts, whereas function words and some relations either become AMR relations or get omitted if they do not contribute to the meaning of a sentence. The `to`, `be`, and `my` words in dependency tree are omitted from the AMR, and the `advmod` relation in the dependency tree becomes the `:manner` relation in the AMR graph. Based on that hypothesis, it is possible to design a set of rules or actions to convert a dependency tree into an AMR structure.

Transition-system based methods convert an input sentence into a corresponding AMR graph, creating an abstract machine characterized by a set of configurations, as a stack of partially processed words, a buffer on unseen input words, and transitions among them.

In order to learn and decide which action to perform, dependency tree and transition-system approaches normally implement a neural network classifier, using Part-Of-Speech (POS) tags, dependency relations, and named entities as features. Furthermore, in the training phase, these approaches require alignment between the word tokens of the sentence and the nodes of the AMR graph for assisting the classifier to learn which token or span tokens is anchored to a node in the AMR graph. In Figure 3, an example of alignment is presented. The alignment

Figure 2 – Dependency tree (left) and AMR graph (right) for the sentence “I like my misfortunes to be taken seriously.” extracted from the Little Prince



Source – (ANCHIÊTA; PARDO, 2018)

format is a space separated list of spans with their graph fragment, where each node is specified by a descriptor: `0` for the root node, `0.0` for the first child of the root node, `0.1` for the second child of the root node and so forth. In this example, the spans `2-3`, `1-2`, and `4-5` that are the tokens `and`, `boy`, and `girl`, respectively, are aligned with the nodes `0`, `0.0`, and `0.1`, which are the root of the graph, and the first and second children of the root node.

Figure 3 – An example of alignment for the sentence “The boy and the girl”

```
::snt The boy and the girl
::alignment 2-3|0 1-2|0.0 4-5|0.1

(a / and
   :op1 (b / boy)
   :op2 (g / girl))
```

Source – Elaborated by the author

For the adoption if a deep learning method, a large annotated corpus is necessary. This approach is align-free and does not need feature engineering to produce an AMR structure. Generally, a *seq2seq* model (SUTSKEVER; VINYALS; LE, 2014) is adopted to learn a linearized AMR structure from an input sentence. Figure 4 shows a linearized AMR structure.

Figure 4 – An example of linearized AMR structure for the sentence “The boy and the girl”

```
(a / and :op1 (b / boy) :op2 (g / girl))
```

Source – Elaborated by the author

To evaluate AMR structures, i.e., to compare the output of an AMR parser against a gold-standard AMR annotation, traditionally, the Smatch metric is used (CAI; KNIGHT, 2013). It uses the conjunction of logical triples to compute precision, recall, and f-score, calculating the degree of overlapping between two AMR structures via one-to-one matching of variables.

1.0.2 Gaps

The AMR language is broadly explored with several corpora[2] and parsers (FLANIGAN *et al.*, 2014; WANG; XUE; PRADHAN, 2015; GOODMAN; VLACHOS; NARADOWSKY, 2016; DAMONTE; COHEN; SATTA, 2017; NOORD; BOS, 2017; LYU; TITOV, 2018; ZHANG *et al.*, 2019a) supported by shared tasks (MAY, 2016; MAY; PRIYADARSHI, 2017; OEPEN *et al.*, 2019) and workshop (XUE *et al.*, 2019). Despite the growing interest in this representation, the researchers focus mainly on the English language, i.e., there are several gaps for the Portuguese language to be filled since there are neither resources nor tools for Portuguese. Furthermore, the available parsers, alignment, and evaluation methods still have some weaknesses, producing unsatisfactory results and revealing the necessity of improvements.

Because of the lack of computationally treatable semantic formalisms, several NLP tasks do not reach satisfactory results yet. For example, automatic summarization and machine translation. The first produces summaries using only surface information (CONDORI; PARDO, 2017), whereas the second uses neural models in an encoder-decoder structure for translating a piece of text. For these two applications, AMR, for example, has improved the summarization (HARDY; VLACHOS, 2018) and achieved competitive results in machine translation (SONG *et al.*, 2019) against the recent neural models supported by word-vector representations.

In this way, one believes that a suitable meaning representation will provide computational tools that reach closer results to those of humans.

1.0.3 Objectives and Hypotheses

The main objective of this work was to study semantic parsing methods, developing, adapting, and evaluating approaches for AMR parsing for the Brazilian Portuguese language. The main hypothesis is that it is possible to map with some accuracy syntactic structures into AMR graphs, applying a set of rules. Although AMR encompasses to semantic level and abstracts way from syntactic phenomena, most of its concepts are lexicalized, that is, they are in the sentence. In this way, AMR allows direct use of the tokens as concepts. Our main hypothesis was confirmed, since it was possible to map the surface textual into AMR structures with some accuracy, producing an AMR parser with strong baseline for the Portuguese language.

Another defined hypothesis is: it is possible to improve existing AMR parsers both adopting a better way of evaluating them in the training phase and enhancing alignment between the tokens of the sentence and the concepts of the graph. Our secondary hypothesis has also confirmed, as the AMR parsers were improved both enhancing alignment between the tokens of the sentence and the nodes of the graph, and using a more robust metric in the training phase of AMR parsers.

[2] <https://amr.isi.edu/download.html>

To achieve the main objective of this work, an exploratory research method and the following specific objectives were adopted:

- Creating an AMR corpus for the development and evaluation of AMR parsing methods.
- Adapting parsing strategies from English (WANG; XUE; PRADHAN, 2015; DAMONTE; COHEN; SATTA, 2017; NOORD; BOS, 2017; DAMONTE; COHEN, 2018) to Portuguese.
- Creating a new specific method for Portuguese that bridges some gaps in the area.
- Dealing with reentrancy relations in the graph and developing a new manner to treat it.
- Evaluating the strongest and weaknesses of the AMR evaluation metric and proposing a new evaluation tool.
- Evaluating alignment methods of the English and developing a new alignment strategy for Portuguese.
- Evaluating the created resources and tools in a Natural Language Processing task.

1.0.4 Outline

In the next chapters, the contributions to the area of Natural Language Processing based on Abstract Meaning Representation are exposed. Each chapter is a paper that was already published or is in the process to be published.

In Chapter 2, some background regarding the AMR formalism, its roots, and some similarities and differences between AMR and other semantic graphs are presented.

In Chapter 3, the process of construction of the first Brazilian AMR corpus is presented (ANCHIÊTA; PARDO, 2018). The proposed align-based method to annotate the corpus is detailed. Besides, an analysis of the annotation phenomena is exposed.

In Chapter 4, the first AMR parser for Portuguese is introduced (ANCHIÊTA; PARDO, 2018). The paper proposes a rule-based method to produce an AMR graph from pre-processed sentences with syntactic and semantic information. The method takes advantage of being align-free. Moreover, the paper extends the Smatch evaluation method analyzing the length of the sentences.

In Chapter 5, a new metric to evaluate AMR structures is shown (ANCHIÊTA; CABEZUDO; PARDO, 2019). The paper introduces the Semantic Evaluation Metric for ARM (SEMA) metric, which extends the Smatch metric. The evaluations show that SEMA is more robust, fairer, and faster than the Smatch metric.

In Chapter 6, a study to evaluate AMR metrics is detailed. The paper details an investigation with humans to find out which AMR metric is more related to human judgment, hence more adequate to evaluate AMR structures.

In Chapter 7, a fine-grained analysis of AMR parsers is presented. In this chapter, a new manner to deal with reentrant relation and the use of the SEMA metric to improve AMR parsers is introduced. Besides, an ablation study with these different settings to show the gain of each of them is performed.

In Chapter 8, an AMR aligner for Portuguese is introduced. This aligner was intrinsically and extrinsically evaluated (ANCHIÊTA; PARDO, 2020b). First, on 100 manually annotated sentences, next on the Brazilian AMR corpus with two adapted AMR parsers. The paper shows that the aligner improved both evaluations.

In Chapter 9, the paper presents a supervised machine learning method, using some semantic features for paraphrasing detection task (ANCHIÊTA; PARDO, 2020a). Besides, the paper explored the potentiality of these features for this task.

In Chapter 10, the contributions to the field are summarized and their impacts and limitations are discussed. Future research directions are addressed as well.

CHAPTER

2

BACKGROUND KNOWLEDGE

In this chapter, some characteristics of the AMR representation, which has its roots in an earlier meaning representation (LANGKILDE; KNIGHT, 1998) are presented in subsection 2.0.1. Besides, the adopted steps to produce or adapt AMR resources for the Portuguese language are shown in subsection 2.0.2.

2.0.1 Aspects of the AMR formalism

Abstract Meaning Representation (AMR) is a semantic graph that arose due to semantic annotation be balkanized. Thus, the AMR formalism proposed to join several semantic annotations as named entities, co-reference, semantic relations, temporal entities, and others into a simple readable SemBank (Semantic Bank) of sentences paired with their whole-sentence and logical meanings (BANARESCU *et al.*, 2013).

In general, semantic graphs may be structurally viewed as directed graphs (or digraphs). A digraph is a pair $G = (V,E)$ where V is a set of vertices (or nodes) and $E \subseteq V \times V$ is a set of directed edges (or arcs) that connect the nodes. In the AMR structure, each node may have multiple outgoing edges and multiple incoming edges (*reentrancies*), except for the root node that not have incoming edges. The latter avoids several instances of the same node. Furthermore, semantic graphs treat nodes as concepts (or events) and edges as relations, as depicted in Figure 1.

Events are the basic building blocks of the predicate-argument structure and, an event encompasses a predicate and arguments. Predicate-argument relations are universally recognized as fundamental to semantic representation (ABEND; RAPPOPORT, 2017). Predicate is the main element that determines what the event is about by evoking elements, while the arguments help to complete the meaning of the predicate. As a predicate-argument structure, some semantic graphs make use of the PropBank framesets (KINGSBURY; PALMER, 2002; PALMER; GILDEA; KINGSBURY, 2005), as the AMR formalism, whereas others make use

of the WordNet senses (MILLER, 1998) and semantic roles by the adapted version of VerbNet roles (KIPPER *et al.*, 2006), as Discourse Representation Graphs (DRG) (KAMP; REYLE, 1993).

In particular, the PropBank (proposition bank) is a resource that was annotated on top of the phrase structure annotation of the Penn TreeBank (MARCUS; SANTORINI; MARCINKIEWICZ, 1993). The PropBank is annotated with verbal propositions and their arguments. It is verb-centered (predicate) in which the arguments of each predicate are annotated with their semantic roles in relation to the predicate (PALMER; GILDEA; KINGSBURY, 2005). In addition to semantic role annotation, PropBank annotation requires the choice of a sense ID, named as a frameset or roleset ID, for each predicate. For example, the sentence "*John opened the door with his foot.*" has one predicate that is the verb **open** with the sense **01**, which means "cause to open" and three arguments, as shown in Figure 5.

Figure 5 – An example of PropBank annotation

Frameset **open.01** "cause to open"

Arg0: agent
Arg1: thing opened
Arg2: instrument

Ex: [Arg0 John] *opened* [Arg1 the door] [Arg2 with his foot]
Source – (PALMER; GILDEA; KINGSBURY, 2005)

One can see the arguments of the verbs are labeled as numbered arguments: Arg0, Arg1, and Arg2. Although numbered arguments correspond slightly different semantic roles given the usage of each predicate, in general, they correspond to the semantic roles as Table 1. From this table, PropBank annotation involves tags to modifiers, as manner, locative, temporal, purpose, cause, and among others. For more details about the PropBank annotation, we suggest consulting the guideline annotation[1].

Table 1 – List of arguments of the PropBank

Argument	Semantic role	Argument	Semantic role
Arg0	agent	Arg3	starting point, benefactive, attribute
Arg1	patient	Arg4	ending point
Arg2	instrument, benefactive, attribute	ArgM	modifier

Another characteristic of semantic graphs is regarding the nature of the relationship they assume between the linguistic surface signal (tokens of a sentence) and the nodes of the

[1] <https://github.com/propbank/propbank-documentation/raw/master/annotation-guidelines/Propbank-Annotation-Guidelines.pdf>

graph (KUHLMANN; OEPEN, 2016). The relationship refers to the alignment (or anchoring) between nodes of the graph and tokens, and it may occur of three manner:

1. A strong anchoring between the tokens of the sentence and the nodes of the graph;
2. A relaxed correspondence between nodes and tokens, allowing arbitrary parts of the sentence as nodes anchors, as well as multiple nodes anchored to overlapping substrings;
3. No anchoring between nodes and tokens.

In the first one, anchoring is obtained in bi-lexical dependency graphs, where graphs nodes injectively correspond to surface lexical units. The Combinatory Categorial Grammar Depencencies (CCG) (HOCKENMAIER; STEEDMAN, 2007) is an example of this semantic graph type. In the second manner, occur a relaxing of the correspondence between nodes and tokens, but still, there is an explicitly annotating of the anchoring between nodes and parts of the sentence. The Universal Conceptual Cognitive Annotation (UCCA) (ABEND; RAPPOPORT, 2013) is an example of a semantic graph with this relaxed alignment. Finally, in the third one, there is no anchoring between nodes and tokens. ARM and DRG are examples of this semantic graph type.

Given these semantic graph types, the more relaxed the alignment between nodes in the graph and tokens in the sentence the less costly for an annotator, making annotation faster and allowing annotators to explore their ideas about how tokens are related to meanings. On the other hand, relaxing the alignment makes the semantic parsing task more challenging, since semantic parsers will have to learn the alignment.

Another effect of the anchorage is the labeling of the nodes in the graph. When the alignment between nodes and tokes is mandatory, node labels tend to be the lemma of the tokens. For semantic graphs with relaxed or no alignment, node labels tend to contain abstract concepts. The AMR formalism, for instance, has 44 abstract concepts[2]. To summarize, Table 2 presents some aspects of the AMR formalism.

Table 2 – Aspects of the AMR formalism

Attribute	**Value**	**Attribute**	**Value**
Event	PropBank	Edge incoming	Multiple
Node label	PropBank frameset, lexicon, special keywords	Edge outgoing	Multiple
Edge label	ArgX, specific AMR relations	Anchoring	None

In the next subsection, steps to produce or adapt AMR resources as corpus, alignment, and parsers for the Brazilian Portuguese language are presented.

[2] <https://amr.isi.edu/doc/amr-dict.html>

2.0.2 AMR formalism for the Portuguese Language

In order to develop or adapt AMR parsing methods for Portuguese, an annotated corpus is required. For that, in Chapter 3, The Little Prince book (written in Brazilian Portuguese) in the AMR formalism was annotated. This book was annotated aiming to compare the similarities and differences concerning its counterpart in English.

Given the annotated resource for Portuguese, a rule-based ARM parsing method was developed and a cross-lingual approach was adapted to Portuguese (Chapter 4). To evaluate and compare these methods, the annotated corpus and the Smatch metric (CAI; KNIGHT, 2013) were used.

Analyzing the weaknesses of the Smatch metric and inspired by (DAMONTE; COHEN; SATTA, 2017), a new evaluation metric named SEMA (Semantic Evaluation Metric for AMR) was developed (Chapter 5). This new metric is stricter than Smatch, as it analyses if the nodes with outgoing edges leading to the target node are into the reference graph.

Aiming to figure out which AMR metric is more consistent with the human judgment, two experiments were carried out (Chapter 6). Based on these investigations, one figured out that SEMA is more consistent with human evaluation and Smatch is less consistent.

To produce AMR parsers more robust for Portuguese, three AMR parsing models from English to Portuguese were adapted (Chapter 7). Moreover, the rule-based AMR parser developed for Portuguese was improved. More than this, taking into consideration that the SEMA metric that produces results more consistent with the human evaluation, one investigated if it improves the adapted AMR parsing models.

Although there is no anchoring in the AMR formalism, most of the parsing models require an alignment to produce the nodes of the graph. Since that the available alignment methods for English have poorly performance for Portuguese, an alignment strategy for Portuguese was developed (Chapter 8).

Finally, in Chapter 9, the developed resources, methods, and tools were evaluated on a paraphrase detection task. For that, a supervised-machine learning strategy to classify if two sentences are paraphrases of each other was adopted.

CHAPTER

3

TOWARDS AMR-BR: A SEMBANK FOR BRAZILIAN PORTUGUESE LANGUAGE

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo. 2018. “Towards AMR-BR: A SemBank for Brazilian Portuguese”. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC).

Contribution statement

R.T. Anchiêta participated in conceiving and designing the annotation process, planned and performed the annotation, and contributed in writing the manuscript. T.A.S. Pardo conceived and designed the annotation process, helped writing the manuscript, and supervised the project.

Towards AMR-BR: A SemBank for Brazilian Portuguese Language

Rafael Torres Anchiêta, Thiago Alexandre Salgueiro Pardo
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo
rta@usp.br, taspardo@icmc.usp.br

Abstract

We present in this paper an effort to build an AMR (Abstract Meaning Representation) annotated corpus (a semantic bank) for Brazilian Portuguese. AMR is a recent and prominent meaning representation with good acceptance and several applications in the Natural Language Processing area. Following what has been done for other languages, and using an alignment-based approach for annotation, we annotated the Little Prince book, which went into the public domain and explored some language-specific annotation issues.

Keywords: Abstract Meaning Representation (AMR), corpus annotation, Portuguese language

1. Introduction

Due to its wide applicability and potentialities, Natural Language Understanding (NLU) has gained interest and fostered research on themes of computational semantics (Oepen et al., 2016). According to Ovchinnikova (2012), NLU is the field of Natural Language Processing (NLP) that deals with machine reading comprehension. The objective of an NLU system is to specify a computational model to interpret one or more input text fragments. The interpretation is usually carried out by a semantic parsing technique, which maps natural language into a suitable meaning representation.

A meaning representation is one of the most important components in semantic parsing. Its production is motivated by the hypothesis that semantics may be used to improve many natural language tasks, such as summarization, question answering, textual entailment, and machine translation, among others. In this context, there are several available meaning representations, as the traditional First-Order Logic (FOL), as detailed in Jurafsky and Martin (2009), semantic networks (Lehmann, 1992), Universal Networking Language (UNL) (Uchida et al., 1996), and, more recently, the Abstract Meaning Representation (AMR) (Banarescu et al., 2013).

In particular, AMR got the attention of the scientific community due to its relatively simpler structure, establishing the connections/relations among nodes/concepts, making them easy to read. Moreover, AMR structures are arguably easier to produce than traditional formal meaning representations (Bos, 2016).

According to Banarescu et al. (2013), AMR-annotated corpora are motivated by the need of providing to the NLP community datasets with embedded annotations related to the traditional tasks of NLP, for instance, named entity recognition, semantic role labeling, word sense disambiguation, and coreference. In this sense, the AMR annotation especially focuses on the predicate-argument structure as defined in PropBank (Kingsbury and Palmer, 2002; Palmer et al., 2005). Another characteristic of AMR annotation is that words that do not significantly contribute to the meaning of a sentence (which are referred as "syntactic sugar" in the original paper) are left out of the annotation, as articles and the infinitive particle "to".

From the currently available datasets, many semantic parsers emerged (Flanigan et al., 2014; Wang et al., 2015; Peng et al., 2015; Goodman et al., 2016; Zhou et al., 2016; Damonte et al., 2017). Furthermore, with the available parsers, some applications were developed for summarization (Liu et al., 2015) and text generation (Pourdamghani et al., 2016; Song et al., 2017), entity linking (Pan et al., 2015; Burns et al., 2016), and question answering (Mitra and Baral, 2016), among others.

Although there are some available annotated corpora, most of them are for English, producing a gap between English and other languages. In addition, creating such corpora is a very expensive task. For instance, Banarescu et al. (2013) took from 7 to 10 minutes to annotate a sentence in AMR representation. However, in spite of the difficulties, it is important to put some effort on corpus creation for other languages. Annotated corpora are important resources, as they provide qualitative and reusable data for building or improving existing parsers, and for serving as benchmarks to compare different approaches.

In order to fulfill this gap, we annotated a corpus in AMR representation for the (Brazilian) Portuguese language, which we report in this paper. In addition, we also detail some differences between Portuguese and English AMR annotations. To the best of our knowledge, this is the first initiative on AMR for Portuguese. We believe that the availability of such a semantic bank[1] in Portuguese will result in new semantic parsers for this language and support the development of more effective NLP applications.

In the following section, we briefly introduce the AMR fundamentals. In Sections 3 and 4, we present our corpus and report the annotation process and its results. Section 5 concludes the paper.

2. Abstract Meaning Representation

Abstract Meaning Representation (AMR) is a semantic representation language designed to capture the meaning of a sentence, abstracting away from elements of the surface syntactic structure, such as part of speech tags, word ordering, and morphosyntactic markers (Banarescu et al., 2013). It may be represented as a single-rooted acyclic directed

[1] A "SemBank", as referred in one of the first AMR papers.

graph with labeled nodes (concepts) and edges (relations) among them in a sentence. AMR concepts are either words (e.g., "girl"), PropBank framesets ("adjust-01"), or special keywords such as "date-entity", "distance-quantity", and "and", among others. PropBank framesets are essentially verbs linked to lists of possible arguments and their semantic roles. In Figure 1, we show a PropBank frameset example. The frameset "**edge.01**", which represents the "move slightly" sense, has six arguments (Arg 0 to 5).

Frameset **edge.01** "move slightly"

Arg0: causer of motion	Arg3: start point
Arg1: thing in motion	Arg4: end point
Arg2: distance moved	Arg5: direction

Ex: [$_{Arg0}$Revenue] *edge* [$_{Arg5}$ up] [$_{Arg2\text{-}EXT}$ 3.4%] [$_{Arg4}$ to $904 million] [$_{Arg3}$ from $874 million] [$_{ArgM\text{-}TMP}$ in last year's third quarter]. (wsj_1210)

Figure 1: A PropBank frameset (Palmer et al., 2005)

For semantic relationships, besides the PropBank semantic roles, AMR adopts approximately 100 additional relations, as general relations (e.g., :mod, :location, :condition, :name, and :polarity), relations for quantities (:quant, :unit, and :scale) and for dates (:day, :month, and :year), among others.

AMR may also be represented in two other notations: in first-order logic or in the PENMAN notation (Matthiessen and Bateman, 1991). For example, Figures 2 and 3 present the canonical form in PENMAN and graph notations, respectively, for the sentences with similar senses in Table 1.

Sentences
The girl made adjustment to the machine.
The girl adjusted the machine.
The machine was adjusted by the girl.

Table 1: Sentences with the same meaning

```
(a / adjust-01
      :ARG0 (g / girl)
      :ARG1 (m / machine))
```

Figure 2: PENMAN notation

AMR assigns the same representation to sentences that have the same basic meaning. Furthermore, as we may observe in the example, the concepts are "adjust-01", "girl", and "machine", and the relations are :ARG0 and :ARG1, represented by labeled directed edges in the graph. In Figure 2, the symbols "a", "g", and "m" are variables, which may be re-used in the annotation, corresponding to reentrancies (multiple incoming edges) in the graph.

Moreover, AMR represents negation in a different way. It uses the *:polarity* relation between the negated concept and the constant '−' (minus signal). For instance, the sentence "I do not much like to take the tone of a moralist.", extracted

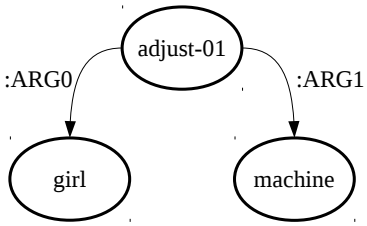


Figure 3: Graph notation

from the Little Prince book, produces the PENMAN notation in Figure 4.

```
(l / like-01 :polarity -
      :ARG0 (i / i)
      :ARG1 (t / take-01
            :ARG0 i
            :ARG1 (t1 / tone
                  :poss (m / moralist)))
      :degree (m1 / much))
```

Figure 4: PENMAN notation representing negation

Finally, to evaluate the AMR structures, Cai and Knight (2013) introduced the Smacth metric, which computes the degree of overlap between two AMR structures, computing precision, recall, and f-score over AMR annotation triples.

3. Our Corpus

There are some available corpora in the Linguistic Data Consortium (LDC), which offer texts in different domains but are not freely available. For now, only two AMR corpora are publicly accessible[2]: Bio AMR Corpus and the Little Prince Corpus. The first includes texts from the biomedical domain, extracted from PubMed[3], whereas the second contains the full text of the famous novel *The Little Prince*, written by Antoine de Saint-Exupéry. The novel was translated into 300 languages and dialects, including Brazilian Portuguese language. Unfortunately, none of the currently available AMR-annotated corpora are for Portuguese.

In this work, following what has been done for other languages, we annotated a public domain version of the Little Prince book written in Portuguese. As a collateral effect of this decision, we may also compare and analyze the annotation of the resulting parallel corpora, composed by the English (source) and Portuguese (target) versions of the book. The original book is organized into twenty-seven chapters. The English version has 1,562 sentences, while the Portuguese one has 1,527. In our annotation process, we aligned all the Portuguese sentences with the English sentences. Furthermore, we calculated some information about the two corpora, such as number of tokens and types, total number of concepts and relations, and maximum and minimum number of concepts and relations found in a sentence, which we show in Table 2.

[2] https://amr.isi.edu/download.html
[3] https://www.ncbi.nlm.nih.gov/pubmed/

Information	English	Portuguese
Number of tokes	16,998	12,703
Number of types	15,829	12,224
Number of concepts	10,528	7,569
Number of relations	10,245	6,676
Average number of tokens	10.88	8.31
Average number of nodes	6	4
Average number of relations	6	4
Maximum number of concepts	37	21
Minimum number of concepts	1	1
Maximum number of relations	49	25
Minimum number of relations	0	0

Table 2: Information about the corpora

4. The Annotation

As aforementioned, we chose as corpus a public domain version of the Little Prince book written in Brazilian Portuguese. Our corpus annotation strategy basically consisted of "importing" the corresponding AMR annotation for each sentence from the English annotated corpus and reviewing the annotation to adapt it to Portuguese characteristics. Doing this, we expected to save time and effort, as a significant part of AMR annotation is probably language independent. More than this, annotation agreement is minimally guaranteed, as it was already checked for the English annotation.

In this sense, we developed an approach with three steps, using the necessary tools and resources to "connect" the English and Portuguese versions of the corpus. Figure 5 illustrates them.

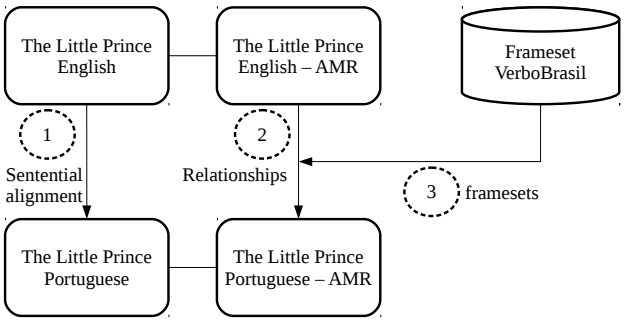


Figure 5: Adaptation of the corpus to the Portuguese language

In the first step, we performed a sentential alignment between the parallel corpora using the TCAlign tool (Caseli and Nunes, 2003), which has a 95% precision. Then, for each sentence, we imported/mapped the AMR relations from the original English sentence to the target Portuguese one. Finally, we included the framesets in each predicate using the *VerboBrasil* dataset (Duran et al., 2013). The *VerboBrasil* dataset is a repository with the sense of verbs in the Portuguese language, similar to the scheme illustrated in Figure 1. This dataset contains examples of a corpus annotated with semantic role labels, created by the PropBank-BR project (Duran and Aluísio, 2012), following the original PropBank initiative. We detail each step in what follows.

Even though the TCAlign tool has 95% precision, we manually checked each alignment, as such information is essential for producing a reliable annotation in Portuguese. We produced 1-1, 1-2, 2-1, 3-1, 1-3, 4-1, 1-4, and 1-5 alignments[4]. As examples, in Tables 3, 4, 5, 6, 7, 8 9, and 10, we present some resulting alignments produced by TCAlign that were manually revised. The overall number for each type of alignment is shown in Table 11. One may also see that there are six sentences in English without correspondence in Portuguese[5].

Source language	Target language
What I need is a sheep.	Preciso é de um carneiro.

Table 3: 1-1 alignment

Source language	Target language
I own three volcanoes, which I clean out every week (for I also clean out the one that is extinct).	Possuo três vulcões que revolvo toda semana.
	Porque revolvo também o que está extinto.

Table 4: 1-2 alignment

Source language	Target language
But I had never drawn a sheep.	Como jamais houvesse desenhado um carneiro, refiz para ele um dos dois únicos desenhos que sabia.
So I drew for him one of the two pictures I had drawn so often.	

Table 5: 2-1 alignment

Source language	Target language
In one of the stars I shall be living.	Quando olhares o céu de noite, porque habitarei uma delas, porque numa delas estarei rindo, então será como se todas as estrelas te rissem!
In one of them I shall be laughing.	
And so it will be as if all the stars were laughing, when you look at the sky at night... you - - only you - - will have stars that can laugh"	

Table 6: 3-1 alignment

In the following steps, we included the sense in each predicate in the sentence, using the *VerboBrasil* dataset, and mapped the relationships to the corresponding AMR relations. Figure 6 shows annotated parallel sentences, in English (left) and in Portuguese (right).

As we see, despite the supposed equality of meaning and annotation, the word '*eu*' (the pronoun "I" in English) does

[4]In an X-Y alignment, X sentences from the original document are aligned to Y sentences in the target one.

[5]Examples of these sentences are "And what good would it do to tell them that?", "Just that.", and "I said.".

Source language	Target language
One sits down on a desert sand dune, sees nothing, hears nothing.	A gente se senta numa duna de areia.
	Não se vê nada.
	Não se escuta nada.

Table 7: 1-3 alignment

Source language	Target language
Hum!.	Hem? respondeu o rei, que consultou inicialmente um grosso calendário.
Hum! ”replied the king; and before saying anything else he consulted a bulky almanac.	
Hum!	
Hum!	

Table 8: 4-1 alignment

Source language	Target language
After that would come the turn of the lamplighters of Russia and the Indies; then those of Africa and Europe, then those of South America; then those of South America; then those of North America.	Vinha a vez dos acendedores de lampiões da Rússia e das Índias.
	Depois os da África e da Europa.
	Depois os da América do Sul.
	Os da América do Norte.

Table 9: 1-4 alignment

Source language	Target language
But in herself alone she is more important than all the hundreds of you other roses: because it is she that I have watered; because it is she that I have put under the glass globe; because it is she that I have sheltered behind the screen; because it is for her that I have killed the caterpillars (except the two or three that we saved to become butterflies); because it is she that I have listened to, when she grumbled, or boasted, or ever sometimes when she said nothing.	Ela sozinha é, porém, mais importante que vós todas, pois foi a ela que eu reguei.
	Foi a ela que pus sob a redoma.
	Foi a ela que abriguei com o pára-vento.
	Foi dela que eu matei as larvas (exceto duas ou três por causa das borboletas).
	Foi a ela que eu escutei queixar-se ou gabar-se, ou mesmo calar-se algumas vezes.

Table 10: 1-5 alignment

not appear in the Portuguese sentence (as it was implicit), but it was annotated. In Portuguese, this phenomenon is called hidden (or implied) subject and it occurs when the subject is not explicit in the sentence but may be easily inferred. In order to keep the similarity with English annotation and the annotation consistency, we annotated all hidden subjects in the Portuguese sentences.

Alignment	Number
1-1	1,356
1-2	41
2-1	60
1-3	3
3-1	10
1-4	1
4-1	1
1-5	1
1-0	6

Table 11: Overall number of alignments

What I need is a sheep	**Preciso é de um carneiro**
(n / need-01	(p / precisar-01
:ARG0 (i / I)	:ARG0 (e / eu)
:ARG1 (s / sheep))	:ARG1 (c / carneiro))

Figure 6: Annotation of parallel sentences

In addition to the subject omission, there are some other differences in the translation into Portuguese. Consequently, the annotation for Portuguese sometimes becomes different from English. In some cases, translations are completely different, such as the one shown in Figure 7. In this example, the owner of the box (poss) and a box modifier (mod) were omitted.

This is only his box	**Esta é a caixa**
(b / box	(c / caixa
:poss (h / he)	:domain (e / esta))
:domain (t / this)	
:mod (o / only))	

Figure 7: An example of translation difference

Other differences are language-specific aspects such as the particle “*se*”, a multifunctional word in Portuguese (which, e.g., may represent the conditional “if” or a reflexive pronoun), words that change their part of speech tags and/or are joined in only one word, and other syntactic features. Figures 8 and 9 illustrate some cases. In Figure 8, one may see that the noun “sweetness” becomes the overall concept “sweet-05”, whereas in Portuguese the overall concept is the verb “*rir-01*” (“to laugh”, in English). Moreover, in Portuguese annotation, it is added the *:manner* relation and the “*docemente*” concept (corresponding to “sweetness”).
In Figure 9, the annotation in Portuguese was very different from the English version. Several concepts and relations were left out in Portuguese annotation, for example, the concepts “contrast-01”, “say-01”, “oh” and the relations “:mod” and “:ARG0-of” were omitted in Portuguese annotation. Moreover, we added the “:poss” relation in Portuguese annotation.
Aiming to organize the number of some of these occurrences/phenomena, we computed and summarized them in Table 12. It is important to notice that the hidden subject phenomenon does not change the original annotation, as we make them explicit. An indeterminate subject, on the other hand, is another type of subject (that may include

And there is sweetness in the laughter of all the stars.

```
(a / and
      :op2 (s / sweet-05
            :ARG1 (l / laugh-01
                  :ARG0 (s1 / star
                        :mod (a2 / all)))))
```

E todas as estrelas riem docemente

```
(e / e
      :op2 (r / rir-01
            :ARG0 (e1 / estrelas
                  :mod (t / todas))
            :manner (d / docemente)))
```

Figure 8: Syntactic structuring variation

But the little prince could not restrain his admiration : " Oh !

```
(c / contrast-01
    :ARG2 (p2 / possible-01 :polarity -
        :ARG1 (r / restrain-01
            :ARG0 (p / prince
                :mod (l / little)
                :ARG0-of (s / say-01
                    :ARG1 (o / oh :mode "expressive")))
            :ARG1 (a / admire-01
                :ARG0 p))))
```

O principezinho, então, não pôde conter o seu espanto

```
(p / poder-201 :polarity -
      :ARG0 (p1 / principezinho)
      :ARG1 (c / conter-02
            :ARG0 p1
            :ARG1 (e / espanto
                  :poss p1)))
```

Figure 9: Syntactic structuring variation

the particle "*se*") that may result in changes in the original annotation. The same happens for some different translations, mainly when they incorporate language specific expressions and constructions.

Phenomenon	#	%
Different translation	494	32.35
Syntactic variation	341	22.33
Hidden subject	285	18.66
Missing verb or sense	191	12.50
Change of predicate	100	6.54
Indeterminate subject	68	4.45
Complex predicate	3	0.19

Table 12: Annotation features in Portuguese

In addition to these phenomena, we calculated the incidence of syntactic variations, changes in predication, and missing verbs or senses. Syntactic variations include part of speech changes, as "little prince" (noun-adjective) to "*principezinho*" (noun), "grown-ups" (noun) to "*pessoas grandes*" (noun-adjective), and "boa constrictor" (noun-noun) to "*jibóia*" (noun), among others. Change of predicate occurs when the predicate in Portuguese is different from English. Thus, this change may produce different arguments.

We also computed the number of included arguments (25) and excluded arguments (103) in relation to English. It is also important to add that *VerboBrasil* is still a small dataset compared to PropBank, and, therefore, did not contain all verbs and senses. In cases where the verbs were not in the dataset, we assigned the sense "01" to the verbs, and marked them in the corpus in order to subsidize future improvements in the *VerboBrasil* repository. These cases occurred 191 times.

A final interesting issue is that importing the AMR structures from the English annotation is helpful, but still demands some effort due to the language specificities. As an illustration, each sentence in Portuguese has 8.31 words in average, and we took about 6 minutes to annotate each one, which is less than the English original annotation from scratch, but is still expensive.

5. Final remarks

The annotated corpus should be made available soon, as the Little Prince book went into public domain. We expect that such annotation may foster research in semantic parsing for Portuguese. Our next steps include to perform wikification of the words, as this also happened for English and looks as a natural step to follow.

More than the annotated corpus availability, our contributions are the proposal of an alignment-based approach for AMR annotation, which we believe that may also be used for other language pairs, and the investigation of annotation issues that may be language specific (in spite of the fact of AMR being a meaning representation).

Acknowledgments

The authors are grateful to FAPESP, CAPES, and *Instituto Federal do Piauí* for supporting this work.

References

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Bos, J. (2016). Expressive power of abstract meaning representations. *Computational Linguistics*, pages 527–535.

Burns, G. A., Hermjakob, U., and Ambite, J. L. (2016). Abstract meaning representations as linked data. In *Proceedings of the 15th International Semantic Web Conference*, pages 12–20.

Cai, S. and Knight, K. (2013). Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the*

51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 748–752.

Caseli, H. and Nunes, M. (2003). Sentence alignment of brazilian portuguese and english parallel texts. In *Proceedings of the Argentine Symposium on Artificial Intelligence*, pages 1–11.

Damonte, M., Cohen, S. B., and Satta, G. (2017). An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546.

Duran, M. S. and Aluísio, S. M. (2012). Propbank-br: a brazilian treebank annotated with semantic role labels. In *Proceedings of the 8th international conference on Language Resources and Evaluation*, pages 1862–1867.

Duran, M. S., Martins, J. P., and Aluísio, S. M. (2013). Um repositório de verbos para a anotação de papéis semânticos disponível na web. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 168–172.

Flanigan, J., Thomson, S., Carbonell, J. G., Dyer, C., and Smith, N. A. (2014). A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1426–1436.

Goodman, J., Vlachos, A., and Naradowsky, J. (2016). Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1–11.

Jurafsky, D. and Martin, J. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.

Kingsbury, P. and Palmer, M. (2002). From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1989–1993.

Lehmann, F. (1992). *Semantic networks in artificial intelligence*. Elsevier Science Inc.

Liu, F., Flanigan, J., Thomson, S., Sadeh, N., and Smith, N. A. (2015). Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086.

Matthiessen, C. and Bateman, J. A. (1991). *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publishers.

Mitra, A. and Baral, C. (2016). Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In *Proceedings of the 30th Conference on Artificial Intelligence*, pages 2779–2785.

Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Cinkova, S., Flickinger, D., Hajic, J., Ivanova, A., and Uresova, Z. (2016). Towards comparability of linguistic graph banks for semantic parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 3991–3995.

Ovchinnikova, E. (2012). *Integration of World Knowledge for Natural Language Understanding*. Atlantis Thinking Machines. Atlantis Press.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Pan, X., Cassidy, T., Hermjakob, U., Ji, H., and Knight, K. (2015). Unsupervised entity linking with abstract meaning representation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1130–1139.

Peng, X., Song, L., and Gildea, D. (2015). A synchronous hyperedge replacement grammar based approach for amr parsing. In *Proceedings of the 9th Conference on Computational Language Learning*, pages 32–41.

Pourdamghani, N., Knight, K., and Hermjakob, U. (2016). Generating english from abstract meaning representations. In *Proceedings of the 9th International Conference on Natural Language Generation*, pages 21–25.

Song, L., Peng, X., Zhang, Y., Wang, Z., and Gildea, D. (2017). Amr-to-text generation with synchronous node replacement grammar. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 7–13.

Uchida, H., Zhu, M., and Della Senta, T. (1996). Unl: Universal networking language–an electronic language for communication, understanding, and collaboration. *Tokyo: UNU/IAS/UNL Center*.

Wang, C., Xue, N., Pradhan, S., and Pradhan, S. (2015). A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.

Zhou, J., Xu, F., Uszkoreit, H., Qu, W., Li, R., and Gu, Y. (2016). Amr parsing with an incremental joint model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 680–689.

CHAPTER

4

A RULE-BASED AMR PARSER FOR PORTUGUESE

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo. 2018. “A Rule-Based AMR Parser for Portuguese”. In Proceedings of the 16th Ibero-American Conference on Artificial Intelligence (IBERAMIA).

Contribution statement

R.T. Anchiêta conceived and developed the research and contributed in writing the manuscript. T.A.S. Pardo conceived the experiments, helped writing the manuscript, and supervised the project.

A Rule-Based AMR Parser for Portuguese

Rafael Torres Anchiêta( ) and Thiago Alexandre Salgueiro Pardo

Interinstitutional Center for Computational Linguistics (NILC),
Institute of Mathematical and Computer Sciences, University of São Paulo,
São Carlos, Brazil
rta@usp.br, taspardo@icmc.usp.br

Abstract. Semantic parsers help to better understand a language and may produce better computer systems. They map natural language statements into meaning representations. Abstract Meaning Representation (AMR) is a new semantic representation designed to capture the meaning of a sentence, representing it as a single rooted acyclic directed graph with labeled nodes (concepts) and edged (relations) among them. Although it is receiving growing attention in the Natural Language Processing community, most of the works have focused on the English language due to the lack of large annotated corpora for other languages. Thus, the task of developing parsers becomes difficult, producing a gap between English and other languages. In this paper, we introduce an approach for a rule-based parser with generic rules in order to overcome this gap. We evaluate the parser on a manually annotated corpus in Portuguese, achieving promising results and outperforming one of the current parser development strategies in the area.

Keywords: Abstract Meaning Representation · Semantic parsing
Portuguese language

1 Introduction

Computational semantics is the area in charge of studying possible semantic representations for human language expressions [14]. A semantic analyzer, also known as a semantic parser, may automatically perform such analysis, and it is responsible for mapping natural language statements into meaning representations, abstracting away from syntactic phenomena and identifying, for example, word senses to eliminate ambiguous interpretations [12]. It aims to understand and translate natural language into a formal meaning representation on which a machine may act, subsidizing more informed and better Natural Language Processing (NLP) systems.

There are several formal meaning representations, as the traditional first-order logic detailed in [14], semantic networks [16], Universal Networking Language [28], and, more recently proposed, the Abstract Meaning Representation

G. R. Simari et al. (Eds.): IBERAMIA 2018, LNAI 11238, pp. 341–353, 2018.
https://doi.org/10.1007/978-3-030-03928-8_28

(AMR) [3], among several others. In special, AMR got the attention of the scientific community due to its relatively simpler structure, showing the relations among concepts and making them easy to read. Moreover, AMR structures are arguably easier to produce than traditional formal meaning representations [6]. At last, AMRs may be evaluated in a standard way by computing precision, recall, and f-measure over gold-standard annotations by the Smatch metric [8].

According to Banarescu et al. [3], AMR was motivated by the need of providing to the research community corpora with embedded annotations related to traditional tasks of NLP, as named entity recognition, semantic role labeling, word sense disambiguation, coreference, and others. From the available corpora, a variety of semantic parsers emerged [10–12,24,32,33], and, with the available parsers, some applications were developed and/or improved: automatic summarization [17], text generation [25,26], entity linking [7,23], and question answering systems [20], for instance.

Most of the parsers are for the English language. However, it is important to develop semantic parsers for other languages in order to support the production of more effective NLP applications. Taking into account the lack of large annotated corpora for non-English languages and the high cost of annotation, semantic parsers based on machine learning approaches become less suitable. Two works tried to overcome these difficulties for non-English languages. Vanderwende et al. [30] developed a set of rules to convert logical forms into AMR representations, and Damonte and Cohen [9] adopted a cross-linguistic approach for creating AMR representations.

In this context, inspired by the above initiatives, in order to create an AMR parser for Portuguese, we developed a rule-based parser. Our parser incorporates a Semantic Role Labeling (SRL) system and a syntactic parser, aiming to preprocess the sentences of interest and producing the respective part of speech tags, dependency trees, named entities, and predicate-argument structures. We then apply a set of manually designed rules on the preprocessed sentences to generate an AMR representation. In addition to the rule-based approach, we adapted for Portuguese the cross-lingual approach of Damonte and Cohen [9] in order to create a baseline system and to compare the results with the rule-based parser. To evaluate these approaches, we adopted a fine-grained strategy introduced by Damonte et al. [10] and we extended it. We noted that the rule-based approach achieved an overall Smatch F-score of 53.5% on the test set, outperforming the cross-lingual approach, which reached 37% of F-score. To the best of our knowledge, this is the first initiative to create an AMR parser for Portuguese.

The remaining of this paper is organized as follows. Section 2 describes the main related work. In Sect. 3, we briefly introduce AMR fundamentals. Section 4 details our rule-based parser. In Sect. 5, we report the experiments and the obtained results. Finally, Sect. 6 presents some conclusions and future directions.

2 Related Work

AMR parsing is a relatively new task, as the AMR language is also new. Several advances have been achieved, but, as the literature review shows us, there is still a long way to go.

Flanigan et al. [11] developed the first AMR parser for English, called JAMR. The authors addressed the problem in two stages: concept identification and relation identification. They handled concept identification as a sequence labeling task and utilized a semi-Markov model to map spans of words in a sentence to concept graph fragments. In the relation identification task, they adopted graph-based techniques of McDonald et al. [19] for non-projective dependency parsing. Instead of finding maximum-scoring trees over words, they proposed an algorithm to find the maximum spanning connected subgraph (MSCG) over concept fragments obtained from the first stage. With this approach, the authors reached a Smatch F-score of 58%.

Wang et al. [32] described a transitional-based parser, named CAMR, that also involves two stages. In the first step, they parse an input sentence into a dependency tree. The second step transforms the dependency tree into an AMR graph by performing a series of manually projected actions. One of the main advantages of this approach is the use of a dependency parser, which may be trained in a large dataset. The CAMR parser obtained a Smatch F-score of 63%. In a posterior work [31], they added a new action to infer abstract concepts and incorporated richer features produced by auxiliary analyzers such as a semantic role labeler and a coreference solver. They reported an improvement of 7% in Smatch F-score.

Peng et al. [24] formalized the AMR parsing as a machine translation problem by learning string-graph/string-tree rules from the annotated data. They applied Markov Chain Monte Carlo (MCMC) algorithms to learn Synchronous Hyperedge Replacement Grammar (SHRG) rules from a forest that represent likely derivations that are consistent with a fixed string-to-graph alignment. They achieved a Smatch F-score of 58%.

Goodman et al. [12] improved the transitional-based parser proposed by Wang et al. [32], applying imitation learning algorithms in order to reduce noise. They achieved a similar performance as that of Wang et al. [31].

Damonte et al. [10] introduced a parser inspired by the `ArcEager` dependency transition system of Nivre [21]. The main difference between them is that Damonte et al. [10] consider the mapping from word tokens to AMR nodes, non-projectivity of AMR structures and re-entrant nodes (multiple incoming edges). They pointed that dependency parsing algorithms with some modifications may be used for AMR parsing. Their parser reached a Smatch F-score of 64%.

The majority of current AMR parsers are for the English language, using some form of supervised machine learning technique that exploits existing AMR corpora. The lack of large annotated corpora for other languages makes the task of developing parsers difficult. To the best of our knowledge, only two works tried to automatically build AMR graphs for non-English sentences. In the first one, Vanderwende et al. [30] produced a parser that may generate AMR graphs

for sentences in French, German, Spanish, and Japanese, where AMR annotations were not available. For this end, they converted logical forms from an existing semantic analyzer [29] into AMR graphs, using a set of rules. In the second approach, Damonte and Cohen [9] proposed a method based on annotation projection, which involves exploiting annotations in a source language and a parallel corpus of the source language and a target language. Using English as the source language, the authors produced AMR graphs in Italian, Spanish, German, and Chinese target languages. Overall, the obtained results are still far from the parsers for English.

3 AMR Fundamentals

Abstract Meaning Representation (AMR) is a semantic representation language designed to capture the meaning of a sentence, abstracting away from elements of the surface syntactic structure such as morphosyntactic information and word ordering [3]. Besides, words that do not contribute to the meaning of a sentence are left out of the annotation. This representation focuses on the predicate-argument structure of a sentence, as defined by the PropBank resource [15,22], and it may be represented as a single-rooted acyclic directed graph with labeled nodes (concepts) and edges (relations) among them. Nodes represent the main events and entities mentioned in a sentence, and edges represent the semantic relationships among nodes.

AMR concepts are either words in their lexicalized forms (e.g., "girl"), PropBank framesets ("adjust-01"), or special keywords such as "date-entity", "distance-quantity", and "and", among others. PropBank framesets are essentially verbs linked to lists of possible arguments and their semantic roles. Figure 1 presents a PropBank frameset example. The frameset "edge.01", whose sense is "move slightly", has six arguments (Arg 0 to 5).

Frameset **edge.01** "move slightly"	
Arg0: causer of motion	Arg3: start point
Arg1: thing in motion	Arg4: end point
Arg2: distance moved	Arg5: direction
Ex: [$_{\text{Arg0}}$Revenue] *edge* [$_{\text{Arg5}}$ up] [$_{\text{Arg2-EXT}}$ 3.4%] [$_{\text{Arg4}}$ to $904 million] [$_{\text{Arg3}}$ from $874 million] [$_{\text{ArgM-TMP}}$ in last year's third quarter]. (wsj_1210)	

Fig. 1. A PropBank frameset [22]

For the semantic relationships, besides the PropBank semantic roles, AMR adopts approximately 100 additional relations. We list below some of them. For more details, we suggest consulting the original paper [3].

– **General semantic relations.** :mod, :location, :manner, :name, :polarity
– **Relations for quantities.** :quant, :unit, :scale
– **Relations for date-entity.** :day, :month, :year, :weekday, :dayperiod
– **Relations for list.** :op1, :op2, :op3, and so on.

In addition to the graph structure, AMR may be represented in two different notations: traditionally, in first-order logic; or in the PENMAN notation [18], for easier human reading and writing. For example, Figs. 2 and 3 present the canonical form in PENMAN and its corresponding graph notation, respectively, for the sentences with similar senses in Table 1.

Table 1. Sentences with similar meaning

Sentences
The girl made adjustment to the machine
The girl adjusted the machine
The machine was adjusted by the girls

```
(a / adjust-01
      :ARG0 (g / girl)
      :ARG1 (m / machine))
```

Fig. 2. PENMAN notation

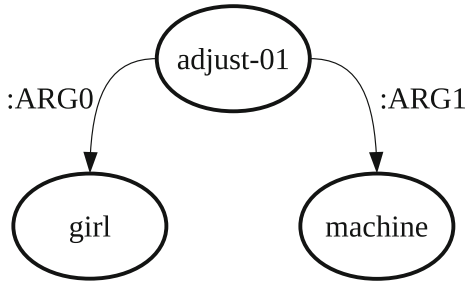


Fig. 3. AMR graph notation

As it is possible to see, AMR assigns the same representation to sentences with the same basic meaning. In the example, the concepts are "adjust-01", "girl", and "machine", and the relations are :ARG0 and :ARG1, represented by labeled and directed edges in the graph. In Fig. 2, the symbols "a", "g", and "m" are variables and may be re-used in the annotation, corresponding to reentrancies (multiple incoming edges) in the graph.

To evaluate AMR structures, Cai and Knight [8] introduced the Smatch metric to asses both inter-annotator agreement and automatic parsing accuracy. This metric computes the degree of overlap between two AMR structures, computing precision, recall, and f-score over AMR annotation triples.

4 A Rule-Based AMR Parser

In order to develop an AMR parser for Portuguese without a large annotated corpus, we designed a set of rules based on dependency links and predicate-argument

structures produced by a syntactic parser and a Semantic Role Labeling (SRL) system, respectively.

We proposed a pipeline organized in three steps: (i) to run a syntactic parser in order to identify the dependency links between the words, morphosyntactic categories, named entities, and the main verb in the sentence; (ii) to execute a SRL tool to extract the predicate-argument structure, and (iii) to apply rules to generate the final AMR. We used the "PALAVRAS" parser [4] and the Brazilis SRL [13], which are state-of-the-art systems for Portuguese.

The syntactic parser produces a dependency structure that has some resemblance with the intended AMR graph. Figure 4 illustrates the similarity between the dependency tree (left) and the AMR graph (right).

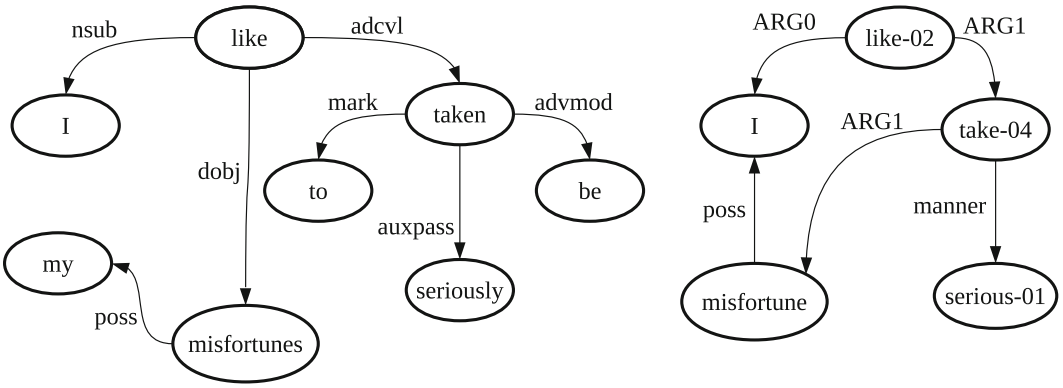


Fig. 4. Dependency tree and AMR graph for the sentence "I like my misfortunes to be taken seriously."

According to Wang et al. [32], in linguistic terms, there are many similarities between the dependency structure of a sentence and an AMR structure. Both describe relations as holding between a parent and its child, or between a head and its dependent. AMR concepts and relations abstract away from actual tokens, but there are regularities in their mappings. Content words generally become concepts, while function words and some relations either become relations or get omitted if they do not contribute to the meaning of a sentence. For instance, 'to', 'be', and 'my' in the dependency tree are omitted from the AMR, and the *advmod* (adverbial modifier) in the dependency tree becomes a *manner* relation in the AMR graph. Furthermore, in AMR, the *poss* relation indicates a reentrancy, used to represent coreference.

After parsing, following the pipeline, the SRL is used to obtain the predicate-argument structure, extensively used by AMR [3]. For the previous sentence, SRL returns the predicates 'like' and 'take' with their respective arguments.

We finally apply a set of rules that were manually developed for the task. Although the AMR has approximately 100 relations, some of them occur more frequently than others and may be produced by our rules. We defined six rules, described below, for the most frequent relations.

- **Named Entity rule.** This rule identifies the named entities indicated by the parser[1] and assigns a concept **name** and their **opn** children. Figure 5 shows the AMR graph for the sentence "At a glance I can distinguish China from Arizona". The parser does not distinguish among country, state, city and other places. It has a unique tag for this, named <*civ*>. Hence, we used ConceptNet [27] to distinguish them.
- **:mod relation rule.** This rule creates a *:mod* relation when an adjective follows a noun[2]. In Fig. 6, we show an AMR example for the sentence "The little prince".
- **:manner relation rule.** This rule applies a *:manner* relation for *advmod* relations of the dependency tree (see Fig. 4).
- **:degree relation rule.** This rule creates a *:degree* relation when the parser produces a relation of adverbial modifier. Figure 7 illustrates this for the sentence "When a mystery is too overpowering".
- **Negative polarity rule.** This rule applies the '−' symbol with the *:polarity* relation when the SRL returns the *AM-NEG* argument. In Fig. 8, we show an example for the sentence "That does not matter".

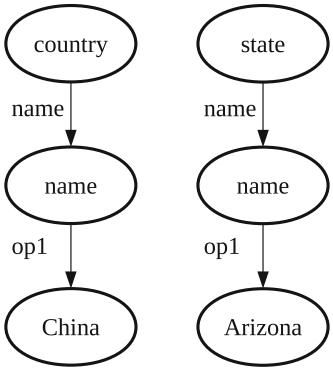


Fig. 5. Rule for named entity

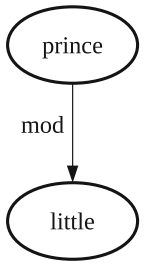


Fig. 6. Rule for :mod relation

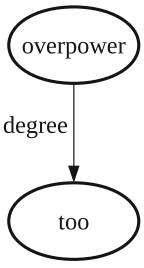


Fig. 7. Rule for :degree relation

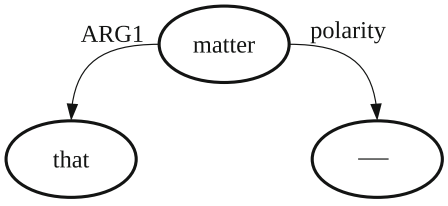


Fig. 8. Rule for negative polarity relation

[1] Although PALAVRAS is a typical syntactical parser, it also produces some shallow semantic annotation.

[2] It is important to notice that this rule was designed for Portuguese, in which the noun-adjective order is the most common ordering.

– **:time relation rule.** This rule creates a *:time* relation when the SRL returns an *AM-TMP* argument. Figure 9 shows an example for the sentence "The little prince said to me later on".

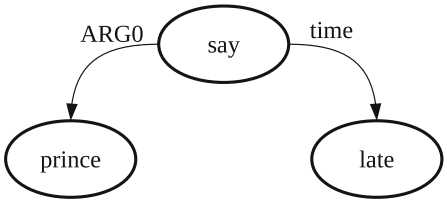


Fig. 9. Rule for :time relation

We designed these rules to be generic, using resources that are common in several languages. For example, the *AM-NEG* and *AM-TMP* arguments are obtained from PropBank, and the *advmod* relation is common in dependency parsers. Thus, we believe that the rules may be reused (with some minor adaptations, if necessary) for other languages without large annotated corpora.

In what follows, we evaluate our semantic parsing strategy.

5 Evaluation

Smatch score [8] is the metric used to evaluate AMR parsers in the area. However, AMR parsing involves many subtasks, as concept identification, named-entity recognition, and negation treatment, among others, and Smatch score consists of single numbers that do not individually assess the quality of each subtask. Therefore, we adopted a fine-grained evaluation introduced by Damonte et al. [10]. More than this, we extended it, analyzing the subtasks by sentence length, as this shows to be an important factor for semantic parsing (the longer the sentence is, the more difficult the semantic parsing is). A fine-grained evaluation shows us the strong points of a semantic parser and, especially, its weaknesses, indicating where we should improve in future work.

As dataset, we used the Little Prince corpus, which was manually annotated for Portuguese [2], keeping the original training/dev/test division proposed for the English version[3]: 1,274, 145, and 143 sentences for training, development, and testing, respectively. Although it may look strange at the first moment, it has been common to use the Little Prince book for AMR processing purposes, as the book went into public domain and had already been adopted by other semantic parsing initiatives that handled different semantic languages.

We computed the average sentence length in the corpus and obtained the 10.46 value. Hence, we organized our evaluation in two ways: for sentences shorter than the average and sentences longer than the average.

[3] https://amr.isi.edu/download.html.

Table 2. F-score results for sentences longer than the average on the test set

Metric	CL (%)	RB (%)
Smatch	29	46
Unlabeled smatch	44	60.5
Concepts	38	61.5
Named entities	43	49
Negations	35	85
# Sentences	80	

Table 3. F-score results for sentences shorter than the average on the test set

Metric	CL (%)	RB (%)
Smatch	45	61
Unlabeled smatch	60	65
Concepts	42	66
Named entities	45	60
Negations	50	88
# Sentences	63	

Table 4. Evaluation for all sentences on the test set

Metric	CL (%)	RB (%)	CL-WA (%)	RB-WA (%)
Smatch	37	53.5	36	52.2
Unlabeled smatch	52	62.7	51	62
Concepts	40	63.7	40	63
Named entities	44	54.5	44	54
Negations	42.5	86.5	42	86
# Sentences	143			

Furthermore, we compared the results of our parser with those of a cross-lingual approach proposed in Damonte et al. [9]. This method is based on word-alignment between two parallel corpora, projecting the AMR structure from the source language (English) to the target (Portuguese) language.

In Tables 2 and 3, we present the F-score results for the test set of the corpus, for longer and shorter sentences, respectively. Table 4 shows the overall average for all sentences and also a weighted average (WA) (as the corpus has different sentence sizes). We show the results for both approaches - the Cross-Lingual (CL) and our Rule-Based (RB) one.

We reported the general results of Smatch and an unlabeled version of it, as well as the fine-grained results for the identification of concepts, named entities and negations. In the unlabeled metric, we only assess the node labels, i.e., we removed all edge labels from the AMR graph. This metric is useful to determine whether two entities are related to each other, not considering the specific type of relationship between them. Concept identification is a critical component of the parsing process: if a concept is incorrectly identified, it is impossible to retrieve any edge involving that concept. We also report results for named entities, which are also related to the concepts and are important to retrieve their related edges. At last, we computed negation detection since it gets researchers special attention [5].

One may see that our rule-based approach achieved better results than the cross-lingual one in all the situations. Specially for shorter sentences, we achieved the best results, as expected (as longer sentences are more prone to error propagation of the syntactic parser and SRL system). Moreover, as AMR is closer to English than other languages, it is less cross-linguistically applicable [1], which may explain the poor results of the cross-lingual approach. As discussed in [2], the Portuguese language shows some differences in relation to the English version of our corpus, as the higher occurrence of hidden subjects, indeterminate subjects, and modifications in part of speech, among others.

We believe that our results are promising given the simplicity of our method, providing a strong baseline for Portuguese. For comparisons purposes, the first AMR parser for English (with better tools and resources than Portuguese) reached a Smatch F-score of 58% and it is used as the baseline for the well-known SemEval tasks, while our first AMR parser for Portuguese presented an overall Smatch F-score of 53.5%. On the other side, one may see that there is a lot of room for improvement. We still have very limited results for identifying concepts, for instance. An error that may be solved by improving the rules is related to the linking verbs. In the sentence "The marble is small", the syntactic parser returns the verb 'to be' as the main verb. However, the verb 'to be' is not used in AMR. In these cases, the root of the graph must be the adjective 'small' instead of the verb 'to be'. Another problem is the generation of duplicate concepts due to the errors of the syntactic parser. For this, pruning methods may be applied to remove duplicate concepts. These improvements may produce better parsing results.

6 Conclusion and Future Work

In this paper, we presented a rule-based AMR parser for Portuguese, trying to overcome the lack of large annotated corpora for system training. We defined a set of generic rules based on the dependency tree relations and the predicate-argument structures from PropBank. We adopted a fine-grained evaluation to verify the performance of the parser and we compared it with a cross-lingual approach. Our parser achieved a Smatch F-score of 53.5%, outperforming the cross-lingual one. To the best of our knowledge, this is the first AMR parsing investigation for Portuguese.

As future work, we intend to improve the set of rules and to test other methods for Portuguese.

Acknowledgments. The authors are grateful to FAPESP and IFPI for supporting this work.

References

1. Abend, O., Rappoport, A.: The state of the art in semantic representation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 77–89 (2017)

2. Anchiêta, R.T., Pardo, T.A.S.: Towards AMR-BR: a semBank for Brazilian Portuguese. In: Proceedings of the 11th Edition of the Language Resources and Evaluation Conference, pp. 974–979 (2018)
3. Banarescu, L., et al.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pp. 178–186 (2013)
4. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus Universitetsforlag, Aarhus (2000)
5. Blanco, E., Moldovan, D.: Semantic representation of negation using focus detection. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 581–589. Association for Computational Linguistics (2011)
6. Bos, J.: Expressive power of abstract meaning representations. Comput. Linguist. **42**, 527–535 (2016)
7. Burns, G.A., Hermjakob, U., Ambite, J.L.: Abstract meaning representations as linked data. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 12–20. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_2
8. Cai, S., Knight, K.: Smatch: an evaluation metric for semantic feature structures. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 748–752 (2013)
9. Damonte, M., Cohen, S.B.: Cross-lingual abstract meaning representation parsing. In: Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies, pp. 1146–1155 (2018)
10. Damonte, M., Cohen, S.B., Satta, G.: An incremental parser for abstract meaning representation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 536–546 (2017)
11. Flanigan, J., Thomson, S., Carbonell, J.G., Dyer, C., Smith, N.A.: A discriminative graph-based parser for the abstract meaning representation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1426–1436 (2014)
12. Goodman, J., Vlachos, A., Naradowsky, J.: Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1–11 (2016)
13. Hartmann, N.S., Duran, M.S., Aluísio, S.M.: Automatic semantic role labeling on non-revised syntactic trees of journalistic texts. In: Silva, J., Ribeiro, R., Quaresma, P., Adami, A., Branco, A. (eds.) PROPOR 2016. LNCS (LNAI), vol. 9727, pp. 202–212. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41552-9_20
14. Jurafsky, D., Martin, J.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, Upper Saddle River (2009)
15. Kingsbury, P., Palmer, M.: From Treebank to Propbank. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation, pp. 1989–1993 (2002)
16. Lehmann, F.: Semantic Networks in Artificial Intelligence. Elsevier Science Inc., Amsterdam (1992)

17. Liu, F., Flanigan, J., Thomson, S., Sadeh, N., Smith, N.A.: Toward abstractive summarization using semantic representations. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1077–1086 (2015)
18. Matthiessen, C., Bateman, J.A.: Text Generation and Systemic-functional Linguistics: Experiences from English and Japanese. Pinter Publishers, London (1991)
19. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 523–530 (2005)
20. Mitra, A., Baral, C.: Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In: Proceedings of the 30th Conference on Artificial Intelligence, pp. 2779–2785 (2016)
21. Nivre, J.: Incrementality in deterministic dependency parsing. In: Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together, pp. 50–57 (2004)
22. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. Comput. Linguist. **31**(1), 71–106 (2005)
23. Pan, X., Cassidy, T., Hermjakob, U., Ji, H., Knight, K.: Unsupervised entity linking with abstract meaning representation. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1130–1139 (2015)
24. Peng, X., Song, L., Gildea, D.: A synchronous hyperedge replacement grammar based approach for AMR parsing. In: Conference on Computational Language Learning, pp. 32–41 (2015)
25. Pourdamghani, N., Knight, K., Hermjakob, U.: Generating English from abstract meaning representations. In: International Conference on Natural Language Generation, pp. 21–25 (2016)
26. Song, L., Peng, X., Zhang, Y., Wang, Z., Gildea, D.: AMR-to-text generation with synchronous node replacement grammar. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 7–13 (2017)
27. Speer, R., Havasi, C.: Representing general relational knowledge in ConceptNet 5. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, pp. 3679–3686 (2012)
28. Uchida, H., Zhu, M., Della Senta, T.: UNL: Universal Networking Language-an Electronic Language for Communication, Understanding, and Collaboration. UNU/IAS/UNL Center, Tokyo (1996)
29. Vanderwende, L.: NLPwin-an introduction. Technical report, Microsoft Research tech report no. MSR-TR-2015-23 (2015)
30. Vanderwende, L., Menezes, A., Quirk, C.: An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus. In: Proceedings of the 2015 Meeting of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies, pp. 26–30 (2015)
31. Wang, C., Xue, N., Pradhan, S.: Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 857–862 (2015)

32. Wang, C., Xue, N., Pradhan, S., Pradhan, S.: A transition-based algorithm for AMR parsing. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 366–375 (2015)
33. Zhou, J., Xu, F., Uszkoreit, H., Qu, W., Li, R., Gu, Y.: AMR parsing with an incremental joint model. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 680–689 (2016)

CHAPTER

5

SEMA: AN EXTENDED SEMANTIC EVALUATION METRIC FOR AMR

Rafael Torres Anchiêta, Marco Antonio Sobrevilla Cabezudo and Thiago Alexandre Salgueiro Pardo. 2019. “SEMA: an Extented Semantic Evaluation Metric for AMR”. Accepted in the Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing).

Contribution statement

R.T. Anchiêta conceived and developed the research and contributed in writing the manuscript. M.A.S. Cabezudo helped in the development of the research. T.A.S. Pardo helped writing the manuscript and supervised the project.

SEMA: an Extended Semantic Evaluation Metric for AMR

Rafael T. Anchiêta[1], Marco A. S. Cabezudo[1], and Thiago A. S. Pardo[1]

Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo, Brazil
rta@usp.br, msobrevillac@usp.br, taspardo@icmc.usp.br

Abstract. Abstract Meaning Representation (AMR) is a recently designed semantic representation language intended to capture the meaning of a sentence, which may be represented as a single-rooted directed acyclic graph with labeled nodes and edges. The automatic evaluation of this structure plays an important role in the development of better systems, as well as for semantic annotation. Despite there is one available metric, *smatch*, it has some drawbacks. For instance, *smatch* creates a self-relation on the root of the graph, has weights for different error types, and does not take into account the dependence of the elements in the AMR structure. With these drawbacks, *smatch* masks several problems of the AMR parsers and distorts the evaluation of the AMRs. In view of this, in this paper, we introduce an extended metric to evaluate AMR parsers, which deals with the drawbacks of the *smatch* metric. Finally, we compare both metrics, using four well-known AMR parsers, and we argue that our metric is more refined, robust, fairer, and faster than *smatch*.

Keywords: Abstract Meaning Representation · Semantic Metric · Evaluation.

1 Introduction

Abstract Meaning Representation (AMR) is a semantic representation language designed to capture the meaning of a whole sentence [4]. AMR got the attention of the scientific community due to its relatively simpler structure, showing the relations among concepts and making them easy to read. The creation of AMR language was motivated by the need of providing to the research community corpora with annotations related to traditional tasks of Natural Language Processing (NLP), such as named entity recognition, semantic role labeling, word sense disambiguation, and coreference resolution [4]. Moreover, AMR structures are arguably easier to produce than traditional formal meaning representations [5].

In this way, several annotated corpora arose, for English[1], Chinese [12], Spanish [15], and Portuguese [3]. Consequently, a considerable number of semantic parsers emerged [9,7,16,2,13], and, with the available parsers, some applications

[1] https://amr.isi.edu/download.html

were developed and/or improved: automatic summarization [10], text generation [18], paraphrase detection [11], and others.

Given the growing interest in AMR language, the automatic evaluation of AMR structures plays a very important role for the AMR parsing task, as well as for semantic annotation tasks, which create linguistic resources for semantic parsing. Although there is one metric to automatically evaluate AMR structures, named *smatch* [6], it has some shortcomings:

1. *Smatch* does not take into account the dependence of the elements in the AMR structure, i.e., its analysis is very simple, masking several analysis problems. So, *smatch* often gives higher scores for AMRs that have different meanings in relation to the reference AMR.
2. *Smatch* creates a self-relation called *TOP* for the root of the AMR structure. That is, *smatch* gives more weight for the root of the graph than other elements, distorting the analysis.
3. *Smatch* has weights for different error types. As discussed by Damonte et al. [7], three named entity errors are considered more important than six wrong labels. Nevertheless, it is difficult to conclude which task should have a higher weight.

Smatch metric computes the degree of overlapping between two AMR structures. To evaluate an AMR generated by a parser against a reference manually produced AMR, *smatch* defines `M` the correct number of triples, `C` the produced number of triples by a parser, and `T` the total number of triples in reference AMR. So, precision and recall are calculated according to Eq. 1 and 2, respectively.

$$P = \frac{M}{C} \qquad (1) \qquad\qquad R = \frac{M}{T} \qquad (2)$$

For example, when evaluating the AMR graph in Fig. 2 against the AMR in Fig. 1, *smatch* returns `M` equal to four (`disaster`, `describe-01`, `man`, and `mission`), `C` equal to eight (`disaster`, `describe-01`, `man`, `mission`, `TOP`, `ARG0`, `ARG1`, and `ARG2`), and `T` equal to eight. So, precision and recall are equal to $4/8 = 0.5$.

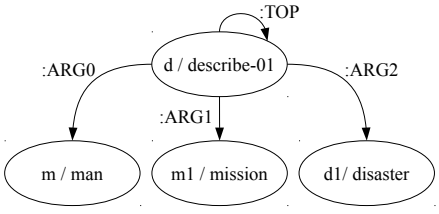


Fig. 1: Reference AMR

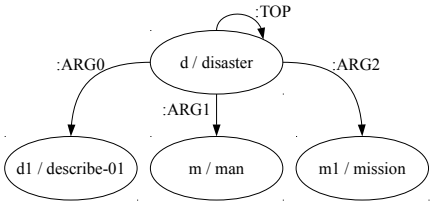


Fig. 2: Test AMR

As we may see, *smatch* adds a *TOP* relation in the structure. This self-relation is not provided by AMR language and it distorts the analysis because if a parser correctly identifies the root of the graph, *smatch* will compute the root concept and the *TOP* relation as correct, otherwise, it will compute only the root concept as correct. In addition, *smatch* is not considering the dependence of the elements. The other issues will be detailed later.

Thereby, we believe that assessing the dependence in which the elements are arranged in the AMR structure may help to better understand the semantic analyzers potentialities and limitations and to produce better applications.

Given these shortcomings and inspired by Damonte et al. [7] to better understand the limitations of AMR parsers and to find their strong points, we propose a new metric for evaluating AMR parsers, named SEMA (Semantic Evaluation Metric for AMR). Our metric deals with these issues of the *smatch* metric, presenting a new way to evaluate concepts and relations in AMR structures, computing precision, recall, and f-score values between two AMRs. Moreover, we compare *smatch* and SEMA, using four well-known AMR parsers in order to analyze the differences between the metrics and, finally, we discuss the obtained results.

In what follows, Sect. 2 presents the essential related work. In Sect. 3, we introduce the AMR fundamentals. Sect. 4 details our developed metric. In Sect. 5, we compare *smatch* and SEMA and, finally, Sect. 6 concludes the paper.

2 Related Work

Compared to traditional meaning representations, AMR is a relatively new representation, as well as AMR parsing is a new task. Thus, there are few works involving semantic representation measurements.

Allen et al. [1] adopted a logical form representation for evaluating its semantic representation. The authors proposed a metric that computes the maximum score by any alignment among logical form graphs. This representation needs an alignment between the input sentences and the semantic analysis. However, the authors did not address how to determine the alignments.

Dridan and Oepen [8] directly evaluated a semantic parser output by comparing semantic sub-structures. The authors also adopted a logical form representation for evaluating its semantic representation. For that, the authors required an alignment between sentence spans and semantic sub-structures. One limitation of that metric is the need for an alignment between the input sentences and their semantic analyses.

Cai and Knight [6] developed a metric named *smatch* that calculates the degree of overlap between two AMR structures. The metric computes the maximum f-score obtainable via one-to-one matching of variables between two AMRs.

As the *smatch* metric, our metric is also focused on AMR structures. However, our metric is more robust, because it deals with the several drawbacks that *smatch* has, as the dependence of elements (nodes, edges), the self-relation

created on the root of the graph, and the weights generated for different error types.

3 AMR Essentials

Abstract Meaning Representation (AMR) is a semantic representation language designed to capture the meaning of a sentence, abstracting away from elements of the surface syntactic structure, such as morphosyntactic information and word ordering [4]. Hence, words that do not significantly contribute to the meaning of a sentence are left out of the annotation.

AMR focuses on the predicate-argument structure of a sentence, as defined by the PropBank resource [17]. It may be represented as a single-rooted directed acyclic graph with labeled nodes (concepts) and edges (relations) among them. Nodes represent the main events and entities mentioned in a sentence, and edges represent semantic relationships among nodes. AMR concepts are either words in their lexicalized forms (e.g., `boy`, `girl`), PropBank framesets (`want-01`, `adjust-01`), or special keywords such as `date-entity`, `distance-entity`, `government-organization`, and others. PropBank framesets are essentially verbs linked to lists of possible arguments and their semantic roles. In Fig. 3, we show a PropBank frameset example. The frameset `edge.01`, which represents the "move slightly" sense, has six arguments (Arg 0 to 5).

Frameset **edge.01** "move slightly"

Arg0: causer of motion	Arg3: start point
Arg1: thing in motion	Arg4: end point
Arg2: distance moved	Arg5: direction

Ex: [$_{Arg0}$Revenue] *edge* [$_{Arg5}$ up] [$_{Arg2\text{-}EXT}$ 3.4%] [$_{Arg4}$ to $904 million] [$_{Arg3}$ from $874 million] [$_{ArgM\text{-}TMP}$ in last year's third quarter]. (wsj_1210)

Fig. 3: A PropBank frameset [17]

For semantic relationships, in addition to PropBank semantic roles, AMR adopts approximately 100 additional relations. We list some of them below. For more details, we suggest consulting the original paper [4].

General semantic relations: :mod, :manner, :location, :name, :polarity
Relations for quantity: :quant, :unit, :scale
Relations for date-entity: :day, :month, :year, :weekday, :dayperiod
Relations for list: :op1, :op2, :op3, and so on

In addition to the graph structure, AMR may be represented in two different notations: traditionally, in first-order logic; or in the PENMAN notation [14], for easier human reading and writing. For instance, Table 1 presents sentences

with similar senses, which are represented in the canonical form in PENMAN format and in the corresponding graph notation, in Figs. 4 and 5, respectively.

Table 1: Sentences with similar meaning

Sentences
The girl made adjustment to the machine.
The girl adjusted the machine.
The machine was adjusted by the girl.

```
(a / adjust-01
      :ARG0 (g / girl)
      :ARG1 (m / machine))
```

Fig. 4: PENMAN notation

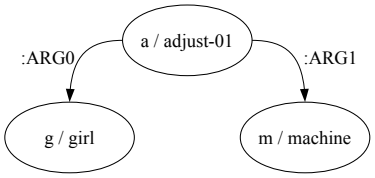


Fig. 5: Graph notation

As it is possible to see, AMR assigns the same representation to sentences with the same basic meaning. In the example, the concepts are `adjust-01`, `girl`, and `machine` and the relations are `:ARG0` and `:ARG1`, represented by labeled directed edges in the graph. In Figs. 4 and 5, the symbols "a", "g", and "m" are variables and may be re-used in the annotation, corresponding to reentrancies (multiple incoming edges) in the graph.

4 SEMA Metric

Following Cai and Knight [6], semantic relationships encoded in the AMR graph may also be viewed as a conjunction of logical propositions, or triples. For example, suppose that the sentence "Tolerance is certainly not fear, and sincerity does not have to be cowardice." produces triples according to Fig. 6 and its graph notation in Fig. 7.

Each AMR triple takes one of these forms: *relation (variable, concept)*, *relation (variable1, variable2)* or *relation (variable, constant)*. The first form encompasses the first seven triples, the second the six triples then, and the third the last two triples in Fig. 6.

Assuming a second AMR annotation for the same sentence, according to Fig. 8 and graphically in Fig. 9, we may compare the two structures considering, for instance, that one is produced by a parser and must be compared to the other one, which would be a reference AMR.

Our metric computes precision, recall, and f-score, evaluating the test triples against the reference triples, analyzing the root of the graphs and, then, relations and concepts, similar to a Breadth-First Search (BFS), taking into account its dependence.

instance (a, and) ^
instance (b, fear) ^
instance (c, certain) ^
instance (d, tolerance) ^
instance (e, obligate-01) ^
instance (f, cowardice) ^
instance (g, sincerity) ^
op1 (a, b)
op2 (a, e)
manner (b, c)
domain (b, d)
ARG2 (e, f)
domain (f, g)
polarity (b, '-')
polarity (e, '-')

Fig. 6: Reference triples

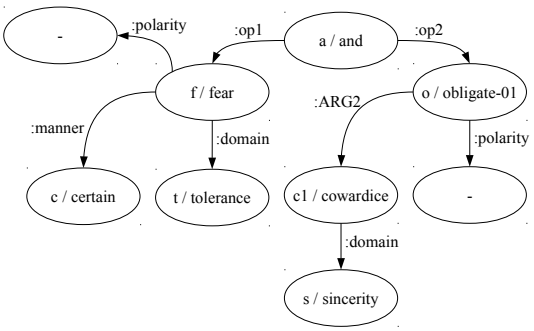


Fig. 7: Graph notation for reference triples

instance (a, and) ^
instance (b, fear-01) ^
instance (c, tolerate-01) ^
instance (d, certain) ^
instance (e, obligate-01) ^
instance (f, cowardice) ^
instance (g, sincerity) ^
op1 (a, b)
op2 (a, e)
ARG0 (b, c)
mod (b, d)
ARG2 (e, f)
ARG1 (e, g)
polarity (b, '-')
polarity (e, '-')

Fig. 8: Test triples

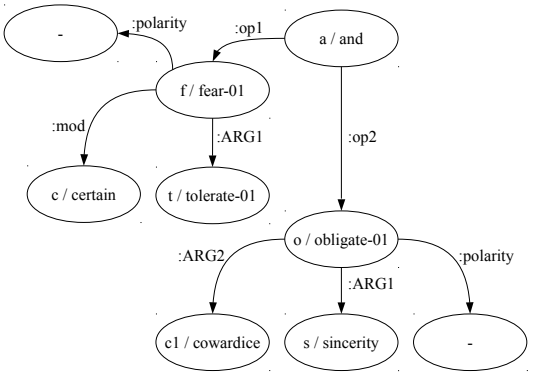


Fig. 9: Graph notation for the test triples

First, our metric analyzes if the root of the test graph (`and`) belongs to the reference graph, that is `and`. We may verify that the two concepts are equal. Thus, the metric computes the concept (`and`) as correct (`M`), one produced element `and` (`C`), and one reference element `and` (`T`). Table 2 presents the root analysis by SEMA.

Table 2: Root analysis

Reference graph	Test graph	M	C	T
and	and	and	and	and

Continuing the evaluation, considering the neighbor relations of the root, our metric analyzes if the relations `:op1` and `:op2` of the test graph and their parent, which is the root of the graph, belong to the reference graph.

Although the two relations are present in reference graph, our metric correctly identifies only the `:op2` relation, as the relation `:op1`, in test graph, is connected to the concept `fear-01` that is different from the reference graph that is `fear`. In Table 3, we show the relations analysis.

Table 3: Relations analysis neighbor to the root

Reference graph	Test graph	M	C	T
:op1, :op2	:op1, :op2	:op2	:op1, :op2	:op1, :op2

After analyzing the relations, our metric analyzes the neighbor concepts of the root, that is, it verifies if the concepts `fear-01`, and `obligate-02` of the test graph and their parent, which is the root of the graph, belong to the reference graph.

Table 4: Concepts analysis

Reference graph	Test graph	M	C	T
fear, obligate-01	fear-01, obligate-01	obligate-01	fear-01, obligate-01	fear, obligate-01

As one may see, the concept `obligate-01` is correct and the concept `fear-01` is wrong, since the correct concept is `fear`. So, the metric computes correctly one element `fear`, shown in Table 4.

In the same manner, our metric will calculate the remaining relations and concepts. At the end of the evaluation, our metric returns six correct triples {*instance (a, and), instance (e, obligate-01), op2 (a, e), instance (f, cowardice), ARG2 (e, f), polarity (e, '-')*} and both test and reference AMRs produced fifteen triples. So, precision, recall, and f-score are equal to $6/15 = 0.40$, respectively.

Analyzing the previous example, the *smatch* metric returns as precision, recall, and f-score values equal to 0.69 for each measure. *Smatch* considers as correct the triples {*instance (a, and), instance (e, obligate-01), instance (d, certain), instance (g, sincerity), instance (f, cowardice), op1 (a, b), op2 (a, e) ARG2 (e, f), polarity (b, '-'), polarity (e, '-'), TOP (a, 'and')*} . The metric tries to maximize the f-score, so, it does not evaluate the dependence of the elements

in the AMR structure. Besides that, the *smatch* scores the root `and` and its self-relation `:TOP`, distorting the analysis[2] (see Fig. 10).

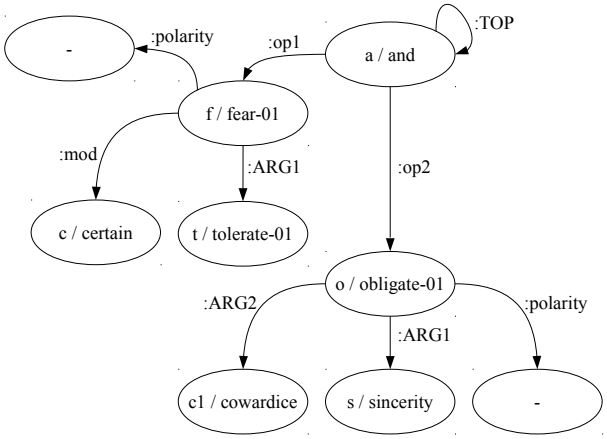


Fig. 10: AMR considered by *smatch* metric

In contrast to the *smatch* metric, our metric considers the dependence of the elements arranged on a graph, i.e., the metric evaluates the relations/concepts and their parents. Furthermore, our metric does not create a `:TOP` relation at the root of the graph, not distorting the evaluation and making the analysis fairer than *smatch* metric. More than that, our metric produces a deterministic result since it works as a Breadth-first search where in the worst-case the performance is $O(|V| + |E|)$, which is faster than to compute the maximum score via one-to-one matching of variable, as the *smatch* metric.

In addition to the above shortcomings, Damonte and Cohen [7] detected that *smatch* has weights for different error types. For example, considering two parses for the sentence "Silvio Berlusconi gave Lucio Stanca his current role of modernizing Italy's bureaucracy", in Fig. 11.

At the left, the output of a parser (*Parser 1*) is not able to deal with named entities. At the right, in the output of other parser (*Parser 2*), except for `:name`, `:op`, and `:wiki` the relation label `:ARG0` is always used. The *smatch* scores for two parses are 0.56 and 0.78 for f-score, respectively. Despite both parses make obvious mistakes, three named entity errors in *Parse 1* are considered more important than six wrong labels in *Parse 2*, according to Damonte et al. [7]. SEMA metric solves that issue by assigning equal weights to all relations, making the evaluation more robust than *smatch*.

[2] The result may be confirmed at `https://amr.isi.edu/eval/smatch/compare.html`. We also checked the available source code `https://github.com/snowblink14/smatch`

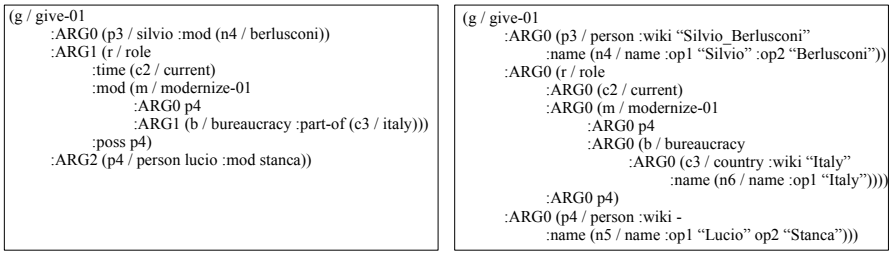


Fig. 11: Sentence "Silvio Berlusconi gave Lucio Stanca his current role of modernizing Italy's bureaucracy" parsed by two parsers [7]

By analyzing AMRs according to SEMA, we may measure precision, recall, and f-score for instance and relation identification tasks, and thus, understand better the AMR parsing task due to a more fine-grained analysis. A demo version and the source code of our metric is available at `https://github.com/rafaelanchieta/sema`. In what follows, we compared our metric with *smatch* using four well-known AMR parsers.

5 Evaluation

In order to compare our metric with *Smatch*, we chose four AMR parsers for English: JAMR parser [9], AMREager parser [7], Neural AMR Parser [16], and AMR Graph Prediction Parser [13]. These parsers were chosen because they handle the parsing task differently and they are publicly available.

We focused on two datasets: LDC2015E86 (R1), which consists of 16,833, 1,368, and 1,371 sentences in training, development, and testing sets, respectively, and LDC2016E25 (R2), which contains 36,521 training sentences, and the same sentences for development and testing as R1. Table 5 shows the comparison between the SEMA and *smatch* metrics on the test set.

Table 5: Comparison between SEMA and Smatch metrics on the test set

Parser	**Train. Data**	**SEMA**			**Smatch**		
		P	R	F	P	R	F
JAMR	R1	0.61	0.57	0.59	0.70	0.64	0.67
AMREAger	R1	0.59	0.54	0.56	0.67	0.62	0.64
Neural AMR	R2	0.67	0.59	0.63	0.76	0.67	0.71
AMR Graph P.	R2	0.67	0.64	0.66	0.75	0.72	0.74

As shown in Table 5, our metric is stricter than *smatch* metric. In order to understand these values and how the metrics deal with graphs of different sizes, we carried out a detailed evaluation.

We calculated the average number of relations in the test set and found that each sentence has 19.8 relations on average. Thus, we organized the test set into two sets: those sentences with number of relations below the average (799 sentences) and those with number of relations above the average (572 sentences) and compared the SEMA and *smatch* metrics. Tables 6 and 7 present the results.

As shown in Tables 6 and 7, in both configurations *smatch* values were superior to SEMA values. This is due to two main factors:

1. The distorted analysis of the relation *TOP*;
2. A large number of concepts and relations not properly evaluated by *smatch*.

In the first factor, in 44.75% of the number of relations below the average and in 77.5% of the number of relations above the average, the parsers did not correctly produce the root of the graph, and, even so, *smatch* considered the roots as correct because the concepts were present in the graph.

Table 6: For number of relation below the average

Parser	Train. Data	SEMA			Smatch		
		P	R	F	P	R	F
JAMR	R1	0.61	0.55	0.58	0.71	0.65	0.68
AMREAger	R1	0.59	0.53	0.56	0.69	0.63	0.66
Neural AMR	R2	0.66	0.62	0.64	0.76	0.72	0.74
AMR Graph P.	R2	0.66	0.64	0.65	0.75	0.73	0.74

Table 7: For Number of relations above the average

Parser	Train. Data	SEMA			Smatch		
		P	R	F	P	R	F
JAMR	R1	0.62	0.58	0.60	0.69	0.64	0.66
AMREAger	R1	0.58	0.54	0.56	0.66	0.61	0.63
Neural AMR	R2	0.68	0.57	0.62	0.74	0.63	0.68
AMR Graph P.	R2	0.68	0.65	0.67	0.75	0.72	0.73

For the second factor, consider the sentence "How long are we going to tolerate Japan?", which was manually annotated as in Fig. 12. The AMR graph has six relations and seven concepts (11 triples). For the same sentence, an AMR parser generated the AMR graph in Fig. 13, which has ten relations and concepts (17 triples).

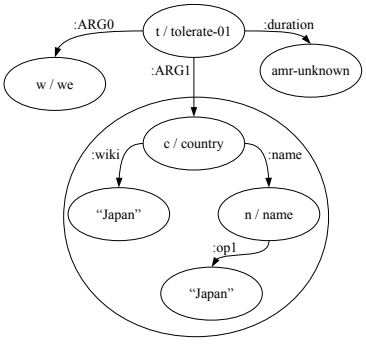


Fig. 12: Reference AMR graph

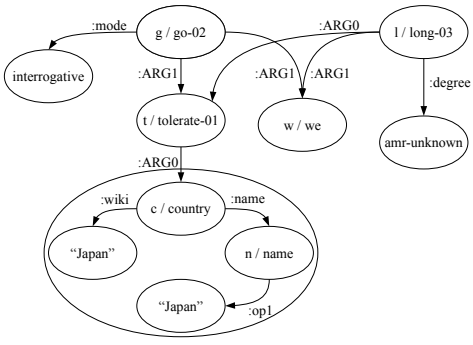


Fig. 13: AMR graph generated by a parser

We may see that the AMR parser produced a subgraph similar to a subgraph that was manually annotated. Despite there are other concepts in the AMR graph produced by the parser that are present in reference AMR graph, as: `tolerate-01`, `we` and `amr-unknown`, their dependents and/or relations are wrong. Hence, the SEMA metric considers these concepts as wrong. For instance, the concept `tolerate-01`, in the reference AMR graph, is the root of the graph, whereas, in the AMR produced by the parser, the root is the concept `go-02`. The root `go-02` is connected to the concept `tolerate-01` through the `:ARG1` relation. Finally, the concept `tolerate-01` in both graphs is connected to the concept `country` but by different relations: `:ARG0` and `:ARG1` for the AMR generated by the parser and reference AMR graph, respectively.

Due to these distinctions, our metric evaluates the connection with the subgraph as wrong since its relation is different from the reference AMR graph. On the other hand, the *smatch* metric evaluates as correct the concepts `we` and `amr-unknown`, although they are not connected to the concept `tolerate-01`. Thus, the *smatch* returns 0.44, 0.67, and 0.53, while the SEMA returns 0.29, 0.45, and 0.36, for precision, recall, and f-score, respectively.

Even though our metric is stricter than *smatch* metric, we believe that SEMA is fairer and more robust than *smatch*. As AMR parsing task is on the semantic level, the dependence of the elements in AMR structure should be analyzed. More than that, SEMA metrics neither creates a TOP self-relation on the root of the graph nor assigns weights for different error types, not distorting the analysis. In the way *smatch* is currently computed, several parsing problems are overlooked.

6 Final Remarks

In this paper, we presented a new metric for evaluating AMR structures. This metric analyzes the dependence in which the elements are arranged in the AMR structure and deals with other shortcomings of the *smatch* metric, as a self-relation produced on the root of the graph, which distorts the analysis, and weights for different error types. We compared our metric with the *smatch* metric, using four AMR parser and showed that, in general, our metric is stricter than *smatch* metric. However, we believe that our metric is fairer and robust than *smatch* since several parsing problems are being overlooked by *smatch*. In addition, we also showed that for both small and large graphs, the parsers have difficulty in learning the dependence of the elements, and even so, *smatch* considers as correct several elements.

As future work, we intend to investigate how to adapt our metric to other semantic representations.

Acknowledgments

The authors are grateful to FAPESP and IFPI for supporting this work.

References

1. Allen, J.F., Swift, M., De Beaumont, W.: Deep semantic analysis of text. In: Proceedings of the 2008 Conference on Semantics in Text Processing. pp. 343–354 (2008)
2. Anchiêta, R.T., Pardo, T.A.S.: A rule-based amr parser for portuguese. In: Simari, G.R., Fermé, E., Gutiérrez Segura, F., Rodríguez Melquiades, J.A. (eds.) Advances in Artificial Intelligence - IBERAMIA 2018. pp. 341–353 (2018)
3. Anchiêta, R.T., Pardo, T.A.S.: Towards amr-br: A sembank for brazilian portuguese. In: Proceedings of the 11th International Conference on Language Resources and Evaluation. pp. 974–979 (2018)
4. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. pp. 178–186 (2013)
5. Bos, J.: Expressive power of abstract meaning representations. Computational Linguistics **42**, 527–535 (2016)
6. Cai, S., Knight, K.: Smatch: an evaluation metric for semantic feature structures. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 748–752 (2013)
7. Damonte, M., Cohen, S.B., Satta, G.: An incremental parser for abstract meaning representation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 536–546 (2017)
8. Dridan, R., Oepen, S.: Parser evaluation using elementary dependency matching. In: Proceedings of the 12th International Conference on Parsing Technologies. pp. 225–230 (2011)
9. Flanigan, J., Thomson, S., Carbonell, J.G., Dyer, C., Smith, N.A.: A discriminative graph-based parser for the abstract meaning representation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. pp. 1426–1436 (2014)
10. Hardy, H., Vlachos, A.: Guided neural language generation for abstractive summarization using abstract meaning representation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 768–773 (2018)
11. Issa, F., Damonte, M., Cohen, S.B., Yan, X., Chang, Y.: Abstract meaning representation for paraphrase detection. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). vol. 1, pp. 442–452 (2018)
12. Li, B., Wen, Y., Weiguang, Q., Bu, L., Xue, N.: Annotating the little prince with chinese amrs. In: Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016. pp. 7–15 (2016)
13. Lyu, C., Titov, I.: Amr parsing as graph prediction with latent alignment. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 397–407 (2018)
14. Matthiessen, C., Bateman, J.A.: Text generation and systemic-functional linguistics: experiences from English and Japanese. Pinter Publishers (1991)
15. Migueles-Abraira, N., Agerri, R., de Ilarraza, A.D.: Annotating Abstract Meaning Representations for Spanish. In: Proceedings of the 11th International Conference on Language Resources and Evaluation. pp. 3074–3078 (2018)

16. van Noord, R., Bos, J.: Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. Computational Linguistics in the Netherlands Journal **7**, 93–108 (2017)
17. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Computational linguistics **31**(1), 71–106 (2005)
18. Song, L., Zhang, Y., Wang, Z., Gildea, D.: A graph-to-sequence model for amr-to-text generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1616–1626 (2018)

CHAPTER

6

THE EVALUATION OF ABSTRACT MEANING REPRESENTATION STRUCTURES

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo. 2018. “The Evaluation of Abstract Meaning Representation Structures”, submitted to the Computational Linguistics journal - Squibs and Discussions, 2019.

Contribution statement

R.T. Anchiêta developed the research, helped in conceiving the evaluation, and contributed in writing the manuscript. T.A.S. Pardo conceived the evaluation, helped writing the manuscript, and supervised the project.

The Evaluation of Abtract Meaning Representation Structures

Rafael T. Anchiêta
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo
São Carlos – SP, Brazil
rta@usp.br

Thiago A. S. Pardo
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo
São Carlos – SP, Brazil
taspardo@icmc.usp.br

Abstract Meaning Representation (AMR) is a sentence-level semantic meaning representation that got the attention of the Natural Language Processing (NLP) community because of its simpler structure, representing a sentence as a direct acyclic graph. This stimulated the development of several AMR corpora and parsers aiming to produce better natural language understanding tools and/or methods and some metrics to evaluate these resources and tools. Smatch is the dominant evaluation metric, but recently SEMA and SemBleu metrics were designed based on the weaknesses of the Smatch. In this paper, we show that the Smatch metric is less suitable to evaluate AMR structures, although it is the most used. We perform an investigation at the sentence and graph levels and found out that SEMA metric is fairer and more adequate to evaluate AMR structures than the other metrics, and the SemBleu metric is fairer than Smatch metric.

1. Introduction

Abstract Meaning Representation (AMR) is a semantic representation language designed to capture the meaning of a sentence (Banarescu et al. 2013). AMR structures may be encoded as a graph with explicit semantic features, such as semantic roles, word sense disambiguation, negation, and other semantic phenomena. In Figure 1 , we present an example of an AMR graph for the sentence "*Something was broken in my engine.*". In this figure, the nodes are concepts and edges are relations among them. The concept `break-01` is the root of the graph and `:ARG1`, `:location`, and `:poss` are AMR relations.

According to Bos (2016), AMR structures are easier to produce than traditional meaning representations. Because of that, AMR corpora for different languages arose, for instance, English[1], Chinese (Li et al. 2016), Portuguese (Anchiêta and Pardo 2018b) and Spanish (Migueles-Abraira, Agerri, and de Ilarraza 2018). With the availability of these corpora, the AMR parsing task, which is responsible for producing an AMR graph from a sentence, got a lot of attention due to the need for better natural language understanding methods (Flanigan et al. 2014; Wang et al. 2015; Artzi, Lee, and Zettlemoyer

1 https://amr.isi.edu/download.html

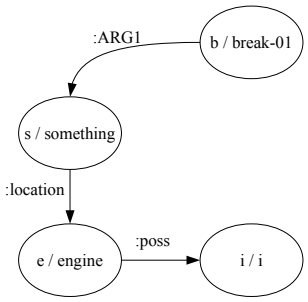


Figure 1
An example of AMR graph for the sentence "*Something was broken in my engine.*" extracted from The Little Prince Corpus.

2015; Damonte, Cohen, and Satta 2017; van Noord and Bos 2017; Anchiêta and Pardo 2018a; Lyu and Titov 2018; Zhang et al. 2019).

The growing interest in AMR language, through the development of several corpora and AMR parsers, stimulated the development of methods/metrics to evaluate these structures, since automatic evaluation plays an important role both in AMR parsing and annotation tasks.

Smatch (Cai and Knight 2013) is the most famous evaluation metric. It measures the degree of overlap between two AMR structures, computing precision, recall, and f-score over AMR annotation triples. Anchiêta, Cabezudo, and Pardo (2019) pointed out that Smatch has two shortcomings. First, it does not take into account the dependency of the elements in the graph, producing a high score for pairs of AMR graphs that are completely different. Second, it adopts a relation named `:TOP` in the root of the graph, distorting the analysis. To deal with these problems, the authors proposed the SEMA metric. Song and Gildea (2019) also indicated two drawbacks in the Smatch metric. First, it is based on a greedy hill-climbing to find a one-to-one node mapping between two AMRs. This produces search errors, weakening the robustness of the metric. Second, it does not take into account the dependency of the elements in the graph, as pointed out by Anchiêta, Cabezudo, and Pardo (2019). To handle these issues, the authors extended the BLEU metric (Papineni et al. 2002) and developed the SemBleu metric for AMR.

Despite these metrics, evaluating AMR structures is not a trivial task, since the metrics may generate very different results for AMR graph pairs. For example, the sentences "*The boy sang very beautifully*" and "*The girl speaks beautifully*" may be encoded as graphs according to Figures 2 and 3, respectively. Comparing the left graph against the right graph, Smatch returns an f-score of 0.43, SEMA gives an f-score of 0.00, and SemBleu yields a value of 0.18. Choosing an evaluation metric that gives over- or under-value results may overlooks several issues, producing unreal values that may mislead the development and improvement of methods and applications.

In this context, based on different similarity values returned by the metrics and the importance of using a fairer metric, we carry out a study aiming to investigate what metric produces results most related to human judgment. Thus, we perform two analysis: at the sentence level and the graph level. In the first one, we create noisy sentences from original ones and ask 5 annotators to rank these sentences from the best (most similar to the original) to the worst sentence (less similar to the original). Next, we use the metrics to create automatic ranks in order to be compared to the ones indicated by the annotators, in order to verify which metric is more suitable for evaluating

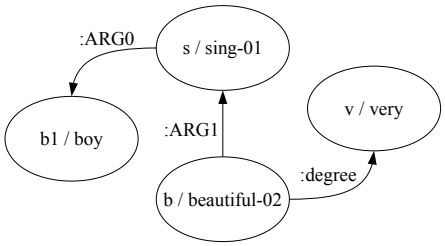


Figure 2
AMR graph for the sentence "*The boy sang very beautifully*"

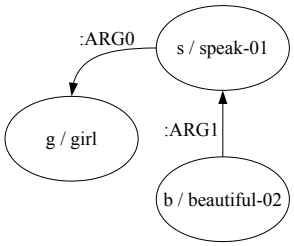


Figure 3
AMR graph for the sentence "*The girl speaks beautifully*"

AMR structures. In the second one, we analyze two AMR properties (inverse relation and reification) to explore their impact on evaluation metrics. Our first analysis takes advantage of the annotators not having to be AMR specialists, i.e., the annotators are responsible only for ranking the noisy sentences concerning the meaning of the original one. We believe that to rank sentences is easier than to analyze and evaluate AMR graphs. In the second analysis, in an opposite direction, we explore specific linguistic phenomena. The first property (inverse relation) allows swapping two AMR relations, inverting a relation, while the second (reification) is the process of turning an AMR relation into a concept. These properties are interesting because they allow to alter the original graph without losing its meaning. Therefore, the metrics should (ideally) return an f-score of 1.0 even though the graphs are not the same. Figures 5 and 6 exemplify the above properties.

The remaining of this paper is organized as follows. In Section 2, we briefly introduce the AMR language. In Section 3, we detail our analysis and the obtained results for the evaluation of the AMR structures. Section 4 concludes the paper and presents future directions.

2. AMR Background

Abstract Meaning Representation (AMR) is a sentence-level semantic representation language that incorporates several semantic features into its structure, for example, semantic role, coreference, named entity and word sense, among others. Words that supposedly do not contribute for the meaning of the analyzed sentence are not annotated, as 'to' infinitive particle and articles, since they are referred to as "syntactic sugar" in the AMR original paper (Banarescu et al. 2013).

As mentioned before, AMR may be encoded as a graph. In addition to graph representation, AMR may be encoded in PENMAN notation (Matthiessen and Bateman 1991) and by a conjunction of logical triples. Figure 4 shows the canonical form in PENMAN and in logical triples for the sentence "*That was by a Turkish astronomer, in 1909.*".

The AMR language has two types of nodes: concrete and abstract (or keywords) nodes. For instance, in the cited figure, the concrete nodes are `see-01` and `astronomer`, since they are present in the text, whereas the abstract nodes are: `country`, `name`, and `date-entity`, as they are not in the text. Besides, these notations have two constants: `Turkey` and `1909`, as they get no variables. AMR semantic relations are `:ARG0`, `:mod`, `:wiki`, `:name`, `:op1`, `:time`, and `:year`. The first AMR relation is a core role and the others are non-core roles. `:mod` indicates a modifier, `:wiki` refers

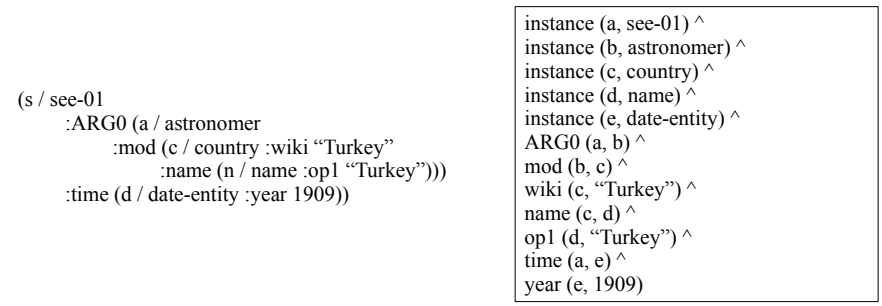


Figure 4
In the left, the PENMAN notation and, in the right, the logical triples

to wikification, `:name` introduces a named entity, `:op1` is used for conjunctions, and `:time` and `:year` are date-entity relations.

The AMR language has two interesting properties: inverse relation and reification. In the first, it is possible to invert two relations without losing the meaning of the sentence, while, in the second, it is possible to turn a relation into a concept. Figure 5 shows an example of inverse relation, where the `:ARG1` relation is converted into `:ARG1-of`, and Figure 6 presents an example of reification, where the `:location` relation is turned into `be-located-at-91` concept.

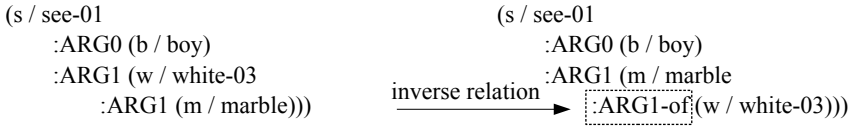


Figure 5
An example of inverse relation for the sentence “*The boy sees that the marble is white*”

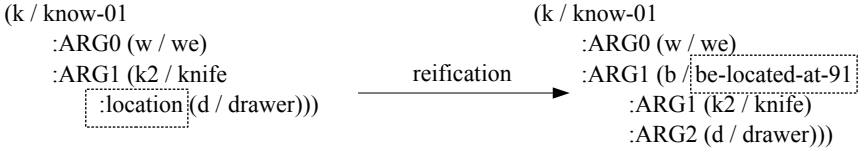


Figure 6
An example of reification for the sentence “*We know the knife is in the drawer.*”

For evaluating AMR structures, the metrics adopt two approaches: conjunction of logical triples or string-match. Smatch and SEMA metrics explore the first strategy, while SemBleu utilizes the second one.

Smatch calculates the degree of overlap between two AMR structures, i.e., it tries to find the one-to-one node mapping between two AMR structures. For computing precision and recall, it defines the following Equations 1 and 2, respectively. M is the correct (according to a reference) number of triples, C is the produced number of predicted triples (e.g., by a parser or a different human annotator), and T is the total number of triples in some AMR reference.

$$P = \frac{M}{C} \quad (1) \qquad R = \frac{M}{T} \quad (2)$$

SEMA adopts the same equations to compute precision and recall. However, instead of finding the one-to-one node mapping, this metric takes into account the dependency (or parents) of the nodes. Thus, it scores nodes and relations only if that condition is satisfied. Another difference is that SEMA removes the `:TOP` relation used by Smatch metric for computing precision and recall.

SemBleu extends the BLEU metric, defining the size of an AMR graph as the number of nodes and edges. This value is used to calculate the brevity penalty (BP) in BLEU, according to Equation 3:

$$BLEU = BP \cdot exp\left(\sum_{k=1}^{n} w_k \log p_k\right) \quad (3)$$

where w_k is the weight for matching k-grams and p_k is the precision. Moreover, SemBleu considers unigrams, bigrams, and trigrams as k-grams for matching and $1/3$ as weight for each n-gram.

To the interested reader, Banarescu et al. (2013) and the AMR guidelines[2] give more details about this language. In what follows, we detail our analysis and results.

3. Analysis and Results

3.1 Sentence level

To evaluate the Smatch, SEMA, and SemBleu metrics, we randomly chose 100 sentences from The Little Prince corpus written in Portuguese (Anchiêta and Pardo 2018b). For each sentence, we created 3 noisy sentences from the original one, and asked 5 annotators to rank these 3 sentences in the best sentence (most similar to the original), worst sentence (less similar to the original), and medium sentence. We chose sentences in Portuguese because the annotators were native speakers of this language.

To create the noisy sentences, we followed a systematic approach, as depicted in Figure 7. In the first sentence, we altered the core predicate of the original sentence[3]; in the second, we changed the arguments of the predicate; and, at last, we removed the adjuncts of the sentence.

From these noisy sentences, we asked the annotators to rank them. In this process, the annotators had agreement greater than or equal to 80% for 70 sentences, and, from these 70 sentences, 80% agreed that the worst sentence is that we changed the core predicate, 85.72% agreed that the best sentence is that we removed the adjuncts, and 72.86% agreed that the medium sentence is that we changed the arguments. Table 1 presents these agreement results. These 70 sentences were the ones we selected for testing the metrics.

2 https://amr.isi.edu/

3 We changed the core predicate by its antonym, as this completely alters the meaning of the sentence.

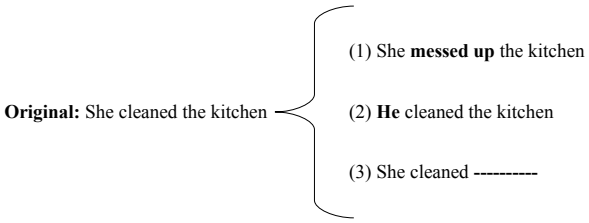


Figure 7
Process to create noisy sentences

Table 1
Agreement between annotators

Noise	Sentence (%)		
	Worst	**Medium**	**Best**
Core predicate	**80.00**	15.71	4.28
Arguments	17.14	**72.86**	10.00
Adjuncts	2.86	11.43	**85.72**

After the annotators ranked the sentences, we manually developed AMR graphs for the noisy sentences. Next, we compared the AMR graphs from the original sentences with the AMR graphs from the noisy sentences, using the Smatch, SEMA, and SemBleu metrics, producing an automatic ranking for each metric from their output scores. In Figure 8, we show this approach. From this figure, for each metric, we computed their output scores comparing the gold AMR graph, at the top, against the AMR graphs at the bottom. Suppose that the annotators ranked the first (**1**), second (**2**), and third (**3**) sentences as worst, medium, and best, respectively. The Smatch metric returned the following f-scores 0.67, 0.83, and 0.80 for the (1), (2), and (3) sentences, respectively. That is, this metric ranked, as worst, best, and medium the three sentences, respectively. SEMA gave 0.00, 0.60, and 0.75 and SemBleu yielded 0.40, 0.57, and 0.51, i.e., they agreed with the manual ranking.

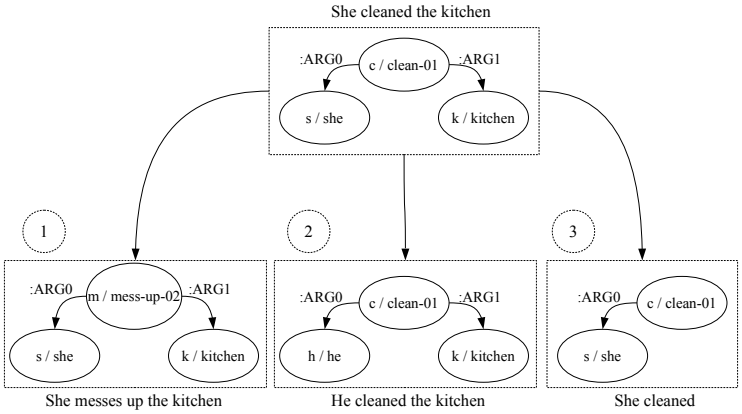


Figure 8
Comparison between the gold AMR graph at the top with the AMR graphs at the bottom

In order to compare the manual ranking with the automatic ones, we used the Kendall's tau coefficient (Kendall 1938), which is a measure of rank correlation. The correlation between two ranks will be 1 when the ranks are identical, and -1 when the ranks are fully different, as defined by Equation 4:

$$\tau = \frac{C - D}{\binom{n}{2}} \tag{4}$$

where C is the number of concordant pairs, D is the number of discordant pairs, and $\binom{n}{2} = \frac{n(n-1)}{2}$ is the binomial coefficient for the number of ways to chose two items from n items. In Figure 9, we show the result of the Kendall's tau correlation for the 70 sentences.

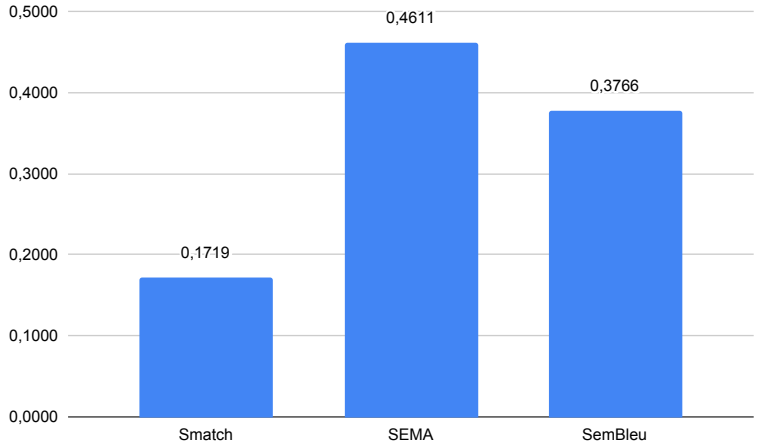


Figure 9
Result of the Kendall's tau correlation

One can see that the ranking produced by the SEMA metric is the most similar to the manual ranking, followed by SemBleu. Surprisingly, Smatch, which currently is the most used metric for AMR, performed very poorly. Therefore, we may consider that SEMA returns fairer and more consistent results in relation to human judgment than the other metrics. Moreover, we checked how many times the metrics agreed with the manual ranking (see Table 2).

Table 2
Agreement result with the manual ranking

Metric	Sentence (%)		
	Worst	Medium	Best
Smatch	47	35	37
SEMA	60	**47**	**60**
SemBleu	**65**	41	45

Looking at Figure 9 and Table 2, SEMA and SemBleu metrics agreed more with the manual ranking. The first one agreed more with the medium and best sentences and

the second one with the worst sentences, i.e., SEMA agreed in 47% and 60% of the 70 sentences with the manual ranking for the medium and best sentences, respectively. The SemBleu metric agreed more with those sentences that we changed the core predicate, that is, 65%, while the SEMA metric agreed more with those sentences that we changed their arguments and removed adjuncts.

3.2 Graph level

At this level, we investigated the inverse relation and reification AMR properties. We chose 10 AMR graphs from The Little Prince corpus, and manually generated a graph version with inverse relation and another with reification, and evaluated these graphs using the metrics. We computed the average of the results of this evaluation and present them in Figure 10.

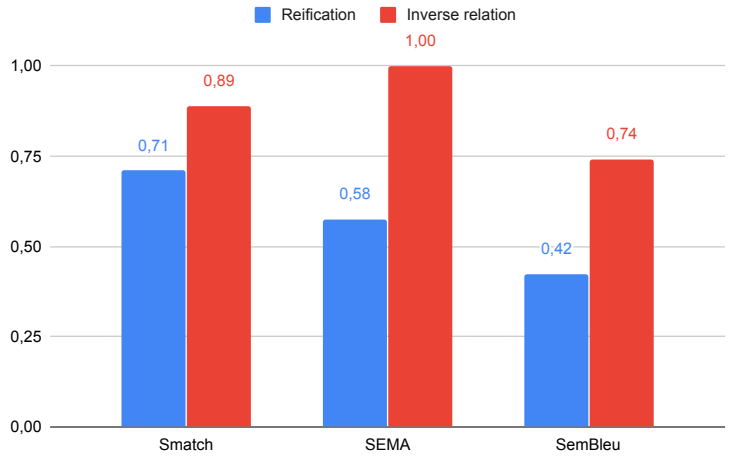


Figure 10
Results for inverse relation and reification properties

We may see that the Smatch metric achieves the better result in reification property. However, we believe that this occurs because the metric yields overvalue results since it has a low correlation with the manual ranking. The SEMA metric reaches an f-score of 1.0 in inverse relation, helping to produce fairer results. However, this metric obtains low results in reification as well as the SemBleu metric. The low values for reification may undervalue AMR parsing and annotation tasks, since an AMR parser may generate a correct AMR graph and be penalized by the metrics (that would evaluate it as a bad quality one). To mitigate the low results in reification, it may be useful for metrics to normalize the AMR structures before evaluating them. Goodman (2019) developed an AMR normalizer to transform a reified AMR into non-reified AMR and vice-versa. We applied this normalizer into our reified version and reassessed the metrics. With this normalized version, Smatch, SEMA, and SemBleu achieved a score of 0.96, 0.94, and 0.89, respectively, improving their scores and making them fairer.

4. Final Remarks

In this paper, we presented an investigation of which AMR evaluation metric produces results that are most related to human judgment. Based on this study, we showed that

the Smatch metric, which is the most famous AMR metric, is less suitable to evaluate AMR structures. We also found out that the SEMA is the more consistent evaluation metric, being more adequate and fairer in relation to human judgment. In addition, incorporating an AMR normalizer as a preprocessing step for evaluation may produce even better results. These findings directly impact the AMR parsing and annotation tasks, since they use AMR metrics to assess their methods/annotations. Adopting a metric that produces over- or under-values may overlook several AMR parsing or annotation issues. More than that, choosing a more consistent and fairer evaluation metric may be useful to find the weakness and strengths of AMR parsing methods. Consequently, the use of the right metric may help developing better applications for natural language understanding.

For future work, we intend to explore a joint approach with SemBleu and SEMA metrics, since the first got a better agreement for sentences less similar to the original, and the second obtained a better agreement for the remaining sentences.

Acknowledgments

The authors are grateful to *Instituto Federal do Piauí* and USP Research Office (PRP 668) for supporting this work.

References

Anchiêta, Rafael Tores and Thiago Alexandre Salgueiro Pardo. 2018a. A rule-based amr parser for portuguese. In *Advances in Artificial Intelligence - IBERAMIA 2018*, pages 341–353.

Anchiêta, Rafael Torres, Marco A. S. Cabezudo, and Thiago Alexandre Salgueiro Pardo. 2019. SEMA: an extended semantic evaluation metric for amr. In *(To appear) Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing*.

Anchiêta, Rafael Torres and Thiago Alexandre Salgueiro Pardo. 2018b. Towards amr-br: A sembank for brazilian portuguese. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, pages 974–979.

Artzi, Yoav, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Bos, Johan. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.

Cai, Shu and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752.

Damonte, Marco, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546.

Flanigan, Jeffrey, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1426–1436.

Goodman, Michael Wayne. 2019. AMR normalization for fairer evaluation. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information, and Computation*.

Kendall, M. G. 1938. A New Measuere of Rank Correlation. *Biometrika*, 30(1-2):81–93.

Li, Bin, Yuan Wen, QU Weiguang, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with chinese amrs. In *Proceedings of the 10th Linguistic Annotation Workshop*, pages 7–15.

Lyu, Chunchuan and Ivan Titov. 2018. Amr parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), volume 1, pages 397–407.

Matthiessen, Christian and John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publishers.

Migueles-Abraira, Noelia, Rodrigo Agerri, and Arantza Diaz de Ilarraza. 2018. Annotating abstract meaning representations for spanish. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 3074–3078.

van Noord, Rik and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Song, Linfeng and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552.

Wang, Chuan, Nianwen Xue, Sameer Pradhan, and Sameer Pradhan. 2015. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.

Zhang, Sheng, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94.

CHAPTER

7

IMPROVING SEMANTIC PARSERS BY A FINE-GRAINED ANALYSIS - THE CASE OF THE PORTUGUESE LANGUAGE

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo. 2018. “Improving Semantic Parsers by a Fine-Grained Analysis - The Case of the Portuguese Language”, submitted to the Language Resources and Evaluation journal, 2019.

Contribution statement

R.T. Anchiêta conceived and developed the research and contributed in writing the manuscript. T.A.S. Pardo helped in conceiving the research and writing the manuscript and supervised the project.

Language Resources and Evaluation manuscript No.
(will be inserted by the editor)

Improving Semantic Parsers by a Fine-Grained Analysis - The Case of the Portuguese Language

Rafael T. Anchiêta · Thiago A. S. Pardo

Received: date / Accepted: date

Abstract Abstract Meaning Representation (AMR) is a recent semantic language 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 Natural Language Processing (NLP) community since 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, improving the adapted models and the rule-based AMR parser. 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 language

Rafael T. Anchiêta
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematial and Computer Sciences
University of São Paulo. São Carlos–SP, Brazil
E-mail: rta@usp.br

Thiago A. S. Pardo
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematial and Computer Sciences
University of São Paulo. São Carlos–SP, Brazil
E-mail: taspardo@icmc.usp.br

1 Introduction

Computational semantics is the area in charge of studying computable semantic representations for human language expressions (Jurafsky and Martin, 2009). In this area, a semantic analyzer, also known as a semantic parser, may automatically perform such task, aiming to understand, translate, or map natural language into a formal meaning representation. This is achieved abstracting away from syntactic phenomena and identifying, for example, word senses, named entities, semantic roles, and others to eliminate ambiguous interpretations on which a machine may act (Goodman et al, 2016). The production of semantic parsers is motivated by the hypothesis that semantics may be used to improve many natural language tasks and produce more informed and better Natural Language Processing (NLP) systems , such as automatic summarization (Liu et al, 2015; Hardy and Vlachos, 2018), text generation (Pourdamghani et al, 2016; Song et al, 2017, 2018), entity linking (Pan et al, 2015; Burns et al, 2016), paraphrase detection (Issa et al, 2018), question answering systems (Mitra and Baral, 2016), and machine translation (Song et al, 2019), among others.

Due to its application, a meaning representation is one of the most important components for a semantic parser. In this way, several formal meaning representations were developed, as the traditional First-Order Logic (FOL) detailed in Jurafsky and Martin (2009), semantic networks (Lehmann, 1992), Universal Networking Language (UNL) (Uchida et al, 1996), Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013), and, more recently, the Abstract Meaning Representation (AMR) (Banarescu et al, 2013).

In special, AMR got the attention of the scientific community due to its relatively simpler structure, showing the relations among concepts and making them easy to read (see Figure 1). The nodes are concepts and edges are relations. The concept `need-01` is the root of the graph and `:ARG0`, `:ARG1`, and `:time` are AMR relations. Moreover, AMR structures are arguably easier to produce than traditional formal meaning representations (Bos, 2016). Also, AMRs may be evaluated in a standard way by computing precision, recall, and f-measure over gold-standard annotations.

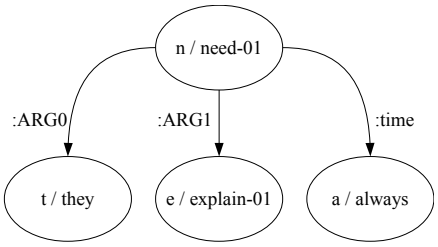


Fig. 1 An example of AMR graph for the sentence "*They always need to have things explained.*" extracted from The Little Prince Corpus.

According to Banarescu et al (2013), AMR was motivated by the need of providing to the research community corpora with embedded annotations related to traditional tasks of NLP, such as named entity recognition, semantic role labeling, word sense disambiguation, coreference, and others. From the available corpora, many AMR parsers with different approaches were developed, for example: based on graph (Flanigan et al, 2014), on dependency-trees (Wang et al, 2015b), on transition-system (Damonte and Cohen, 2018), deep learning (van Noord and Bos, 2017; Lyu and Titov, 2018), and others.

Most of the AMR parsers are available only for English due to the lack of large annotated corpora in other languages. This lack produces a gap of resources and tools between English and other languages. Because of this gap and aiming to support the production of more effective NLP applications, it is interesting to develop and/or adapt semantic parsers for other languages. Following this strategy, Wang et al (2018) adapted a semantic parser (Wang et al, 2015b) based on dependency-trees for Chinese, producing the first AMR parser for that language.

In this context, in this paper, we adapted some available AMR parsers from English to Portuguese. Although several authors have proposed different approaches for AMR parsers, there are a lot of rooms for improvements. For example, most of these parsers adopt a supervised machine learning strategy that requires alignment among tokens of a sentence and nodes of an AMR graph, and the alignment has some shortcomings (as re-entrants in the graph). Thus, we performed a fine-grained analysis these parsers and identified some gaps. Aiming to overcome them, we proposed some improvements. Together with the adapted AMR parsers, we improved a Portuguese designed AMR parser.

In order to analyze the performance of the parsers, we carry out an evaluation using the traditional metric, named Smatch (Cai and Knight, 2013), and a new metric, named SEMA (Anchiêta et al, 2019), which is stricter, faster, and more robust than Smatch, on a manually annotated corpus in Portuguese (Anchiêta and Pardo, 2018b). Then, we perform a detailed error analysis to identify the major challenges in Portuguese AMR parsing, hoping that it will inform and foster future research in this area.

In general, this paper makes the following contributions: (i) adaptation and improvement of AMR parsing models from English to Portuguese, (ii) improving a Portuguese AMR parser, (iii) an overview of AMR parsing methods, and (iv) a detailed error analysis of AMR parsers.

The remaining of this paper is organized as follows: in Section 2, we briefly introduce the main concepts regarding AMR language; Section 3 presents the available corpora in this area, some main related work and approaches to produce an AMR graph; Section 4 details the adapted parsers and implemented improvements; in Section 5, we report the experiments and the obtained results; finally, in Section 6, we present our conclusions and future directions.

2 AMR Essentials

Abstract Meaning Representation (AMR) is a semantic representation language designed to capture the meaning of a sentence, abstracting away from elements of the surface syntactic structure such as morphosyntactic information and word ordering (Banarescu et al, 2013). AMR language discards the words that do not contribute to the meaning of the sentence, as articles and the infinitive particle "to". It also neglects quantifiers, grammatical number, tense, aspects, and others to simplify the representation. Furthermore, it focuses on the predicate-argument structure of a sentence, as defined by the PropBank resource (Kingsbury and Palmer, 2002; Palmer et al, 2005).

AMR may be represented as a single-rooted directed acyclic graph with labeled nodes (concepts) and edges (relations) among them, where nodes represent the main events and entities mentioned in a sentence, and edges represent the semantic relationships among nodes. AMR concepts may be concrete, which encompasses words in their lexicalized forms (e.g. "woman") and PropBank framesets ("want-01"), or abstract (or special keywords) that do not correspond to any lexical unit, such as "email-address-entity", "percentage-entity", and "distance-quantity", among others.

AMR concepts that are PropBank framesets are normally verbs linked to lists of possible arguments and their semantic roles. Figure 2 presents an example where the frameset "fall.01", whose sense is "move downward", has four arguments (Arg 1 to 4).

Frameset **fall.01** "move downward"	
Arg1: Logical subject, patient, thing falling	Arg3: start point
Arg2: EXT, amount fallen	Arg4: end point
Ex: Profits fell 10% to $118 million from $ 130.6 million	
Arg1: Profits	Arg3: to $118 million
Arg2: 10%	Arg4: from $130.6 million

Fig. 2 A PropBank frameset

For AMR relations, besides the PropBank semantic roles (which are the core of AMR relations) approximately 100 additional relations are used. We list below some of them, and, for more details, we suggest consulting the original paper (Banarescu et al, 2013) and/or annotation guidelines[1].

- **General semantic relations.** :mod, :location, :manner, :name, :polarity
- **Relations for quantities.** :quant, :unit, :scale
- **Relations for date-entity.** :day, :month, :year, :weekday, :dayperiod
- **Relations for list.** :op1, :op2, :op3, and so on

[1] `https://github.com/amrisi/amr-guidelines/blob/master/amr.md`

In addition to the graph structure, AMR may be represented in different notations: traditionally, in first-order logic; or in the PENMAN notation (Matthiessen and Bateman, 1991), for easier human reading and writing. For example, Figure 3 presents the canonical form in PENMAN (left) and its corresponding graph notation (right), respectively, and Figure 4 shows the conjunction of logical triples for the sentences with similar senses in Table 1.

Table 1 Sentences with similar meaning

Sentences
The boy did not see the girl who wanted him.
The boy did not see the girl who he was wanted by.
The girl who wanted the boy was not seen by him.

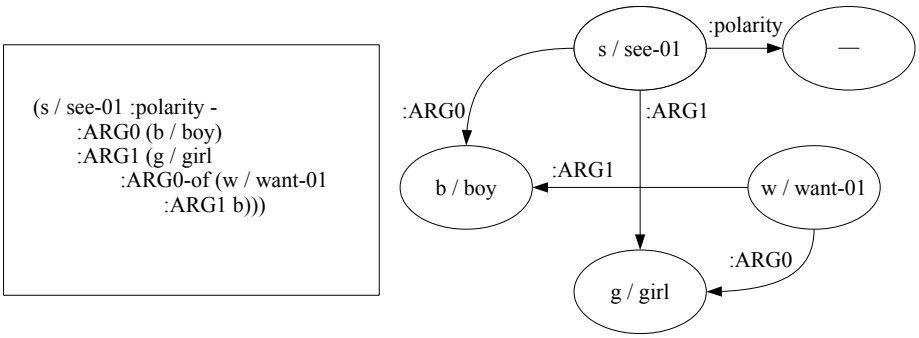


Fig. 3 PENMAN notation (left) and AMR graph (right). In this example, "see-01", "boy", "girl", and "want-01" are concepts, ":ARG0" and ":ARG1" are relations, ":polarity -" is a constant, ":ARG0-of" is an inverse relation, "s", "b", "g", and "w" are variables and, at last, the variable "b" is also used a reentrancy (multiple incoming edges) to avoid repeating the concept "boy".

For evaluating AMR structures, traditionally, the authors use Smatch metric (Cai and Knight, 2013). This metric computes the degree of overlap between two AMR structures, computing precision, recall, and f-score over AMR annotation triples. More recently, Anchiêta et al (2019) developed a new metric to evaluate AMR structures, named Semantic Evaluation Metric for AMR (SEMA) . This metric is stricter than Smatch metric, as it takes into account the dependency in the AMR structures. Moreover, it deals with some shortcomings of Smatch, being faster and more robust than the Smatch metric. Furthermore, these metrics may be useful for semantic annotation tasks, which are responsible for creating linguistic resources for semantic parsing. In the next section, we present the available AMR corpora.

```
instance (s, see-01) ^
instance (b, boy) ^
instance (g, girl) ^
instance (w, want-01) ^
polarity (s, -) ^
ARG0 (s, b) ^
ARG1 (s, g) ^
ARG1 (w, b) ^
ARG0 (w, b)
```

Fig. 4 Example of logical triple notation. Each triple takes one of these forms: *relation (variable, concept), relation (variable, constant) or relation (variable1, variable2)*. The first form encompasses the four first triples, the second the fifth triple, and the third the last four triples.

3 Resources and Related Work

3.1 AMR Corpora

There are several available AMR corpora for English and some initiatives for non-English language. Here, we introduce these corpora.

For English, human annotators manually constructed various corpora and the Linguistic Data Consortium (LDC) is responsible for releasing them. The text of these corpora is in different domains, such as newswire, discussion forums and other weblogs. In Table 2, we detail how these corpora are organized and the total number of sentences for each corpus.

Table 2 AMR Corpora

Language	Available	Corpus	Training	Development	Testing	Total
English	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 LDC2017T10	36,521	1,368	1,371	39,260
English	Public	The Little Prince	1,274	145	143	1,562
		Bio AMR	5,452	500	500	6,452
Chinese	Public	The Little Prince	1,274	145	143	1,562
Portuguese	Public	The Little Prince	1,274	145	143	1,562
Spanish	Public	The Little Prince		50		

The LDC2015E86, LDC2016E25, and LDC2017T10 corpora have the same sentences for development and testing, and the LDC2016E25 and LDC2017T10 corpora are equal[2]. The scientific community broadly uses these resources, although they are not publicly available. For now, there are only two AMR

[2] The LDC2017T10 is available to all LDC subscribers, while LDC2016E25 is limited to DEFT participants. In this paper, we will keep the name of the corpus used in the original work.

corpora for English with that feature[3]: The Little Prince Corpus and Bio AMR Corpus. The first contains the full text of the famous novel *The Little Prince*, written by Antoine de Saint-Exupéry, published in 1943 and translated into 300 languages, whereas the second includes texts from the biomedical domain, extracted from PubMed[4].

For non-English languages, there are some initiatives in AMR annotation tasks, such as in Chinese (Li et al, 2016), Portuguese (Anchiêta and Pardo, 2018b), and Spanish (Migueles-Abraira et al, 2018). These initiatives annotated a version of *The Little Prince* in their respective languages. However, for Spanish, the authors annotated only 50 sentences. It is important to say that *The Little Prince* corpus for Portuguese has only 1,527 sentences and the authors aligned these sentences with their counterparts in English. For more details about the annotation process, we suggest consulting the original papers.

3.2 Related Work

Although AMR parsing is a relatively new task, as the AMR language is also recent, several studies were proposed to produce an AMR graph. Here, we present the main works categorized into five classes: graph, tree, transition-system, Combinatory Categorial Graph (CCG), and deep learning. Moreover, we organized this section in three subsections: parsers for English (3.3), non-English (3.4), and a summary regarding these parsers (3.5).

3.3 Parsers for the English language

3.3.1 Graph-based methods

The graph-based methods identify the nodes and then compute scores of edges, adopting a maximum spanning connected subgraph algorithm to produce edges that will connect the nodes and comprise a graph.

Flanigan et al (2014) developed the first AMR parser for English, called JAMR. The authors addressed the task into two stages: concept identification and relation identification. They handled concept identification as a sequence labeling task and adopted a semi-Markov model to map spans of words in a sentence to concept graph fragments. In the relation identification step, they proposed an algorithm to find the maximum spanning connected subgraph over concept fragments obtained from the first stage. For training these steps, the authors developed an aligner to map the tokens of a sentence with the nodes of a graph. In this way, the authors reached a Smatch f-score of 58% on part of the LDC2013E117 corpus.

[3] `https://amr.isi.edu/download.html`

[4] `https://www.ncbi.nlm.nih.gov/pubmed/`

Werling et al (2015) adopted the work of Flanigan et al (2014) as a basis and proposed a the concept identification task, since 38% of the words in the LDC2013E117 development set are unseen during training time, making memorization-based approaches brittle. The authors evaluated their approach on the LDC2014T12 corpus and on part of the LDC2013E117 corpus, achieving a Smatch f-score of 62.2% and 63.3%, respectively.

3.3.2 Tree-based methods

The tree approaches start from a dependency tree and incrementally modify it into an AMR structure.

Wang et al (2015b) created a parser named CAMR that involves two stages. In the first step, it parses an input sentence into a dependency tree, whereas the second step transforms the dependency tree into an AMR graph by performing a series of manually projected actions. For example, one of these actions is to transform the preposition `in` into the `:location` AMR relation. One of the main advantages of this approach is the use of a dependency parser, which may be trained in a large dataset. The CAMR parser obtained a Smatch f-score of 63% on part of the LDC2013E117 corpus. In a posterior work Wang et al (2015a), added a new action to infer abstract concepts and incorporated richer features produced by auxiliary analyzers, such as a semantic role labeler and a coreference solver. They reported an improvement of 7% in Smatch f-score.

Goodman et al (2016) improved the parser proposed by Wang et al (2015b), applying imitation learning algorithms in order to reduce noises. With this, they achieved similar performance as that of Wang et al (2015a) on part of the LDC2013E117 corpus.

3.3.3 Transition-based methods

A transition system is an abstract machine characterized by a set of configuration (stack of partially processed words and a buffer on unseen input words) and transitions between them. This approach converts an input text sentence into a corresponding AMR graph.

Zhou et al (2016) proposed a transition system in order to alleviate error propagation in the traditional pipeline by performing the concept and relation identification tasks jointly in an incremental model. Their model relies on a set of produced alignments using lexical cues and hand-written rules. In this way, they obtained a Smatch f-score of 67% on the LDC2014T12 corpus.

Damonte et al (2017) introduced a parser inspired by the `ArcEager` dependency transition system of Nivre (2004). The main difference between them is that the first considers the mapping from word tokens to AMR nodes, non-projectivity of AMR structures, and re-entrant nodes (multiple incoming edges). Projectivity is related to the non-crossing condition. If we draw edges in a semi-plane above the words and there are no crossing arcs, the structure is projective, otherwise it is non-projective (Kuhlmann and Jonsson, 2015) (see Figure 5).

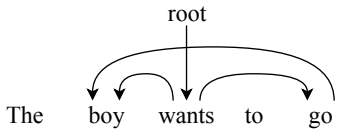


Fig. 5 An example of non-projectivity for the sentence "*The boy wants to go*". The edge connecting the word '*go*' to the word '*boy*' crosses another edge.

The authors pointed that dependency parsing algorithms with some modifications may be used for AMR parsing. Their parser reached a Smatch f-score of 64% on the LDC2014T12 corpus.

Ballesteros and Al-Onaizan (2017) introduced stack-LSTMs to represent the parser state and greedily make decisions for each transition. Moreover, the authors added information, such as Part-Of-Speech (POS) tags and dependency trees, to improve their results. They evaluated their approach on part of the LDC2013E117 and LDC2014T12 corpora and achieved a Smatch f-score of 69% and 64%, respectively.

Peng et al (2018) presented a transition system that generalizes transition-based dependency parsing techniques to generate AMR graphs. For that, they used a fixed-size cache, allowing the system to build arcs to any vertices present in the cache at the same time. The author evaluated their approach on the LDC2015E86 corpus and reached a Smatch f-score of 64%.

Vilares and Gómez-Rodríguez (2018) explored unrestricted non-projective AMR parsing and introduced `AMR-COVINGTON` inspired by Covington (2001). It handles arbitrary non-projectivity, cycles[5] and reentrancy in a natural way, as there is no need for specific transitions, but just the removal of restrictions from the original algorithm. With this approach, the authors reached a Smatch f-score of 64% on the LDC2015E86 corpus.

Guo and Lu (2018) proposed a refined search in the transition-system actions. For that, they designed a representation called *compact AMR graph* to simplify concepts and relations of an AMR graph, which makes the learning of their transition-system easier. In addition, the authors improved the alignments produced by JAMR in order to improve the oracle algorithm. So, they evaluated their approach on part of the LDC2013E117 corpus and on the LDC2014T12, LDC2015E86, and LDC2017T10 corpora, reaching a Smatch f-score of 74%, 68.3%, 68.7%, and 69.8%, respectively.

3.3.4 CCG-based methods

Combinatory Categorial Grammar (CCG) is a categorial formalism that provides a transparent interface between syntax and semantics (Steedman, 1996, 2001).

Artzi et al (2015) mapped sentences to AMR structures in a two stage process. First, they adopted a CCG system to construct lambda-calculus representations of compositional aspects of AMR. Then, the authors proposed a

[5] According to AMR guidelines, approximately 0.3% of AMRs are legitimately cyclic.

CCG grammar induction algorithm to produce an AMR graph. They achieved a Smatch f-score of 66.1% on part of the LDC2013E117 corpus.

Misra and Artzi (2016) developed a CCG semantic parser that uses a neural network architecture, where each parsing step is treated as a multi-class classification problem. In addition, they proposed an iterative algorithm that automatically selects the best parses for training at each iteration, and identifies partial derivations for best-effort learning, if no parses are available. They evaluated this approach on part of the LDC2014T12 corpus and reached a Smatch f-score of 66.1%.

3.3.5 Deep Learning-based methods

Deep Learning models learn to produce an AMR graph from the AMR annotated corpus without feature engineering. Most of the works adopt a *Seq2seq* model that converts an input text into an AMR graph.

Peng et al (2017) adopted a sequence-to-sequence model based on work of Vinyals et al (2015) and proposed AMR linearization[6] and categorization to avoid data sparsity. Thus, the authors achieved 52% of f-score on the LCD2015E86 corpus.

Konstas et al (2017), instead of performing an AMR categorization, such as Peng et al (2017), used 20M unlabeled Gigaword sentences for paired training. In this way, they reached a Smatch f-score of 62.1% on the LDC2015E86 corpus.

van Noord and Bos (2017) adopted a similar approach of Konstas et al (2017). They used an additional training corpus of 100k sentences (called "silver") generated by an ensemble system consisting of JAMR and CAMR parsers. They achieved a Smatch f-score of 71% on the LDC2016E25 corpus.

In addition to *seq2seq* model, other studies addressed the AMR parsing task using several (Long Short-Term Memory) LSTM models.

Foland and Martin (2017) used multiple bidirectional LSTMs to identify different types of concepts and relations and iteratively combine these components to form the AMR graph. Their system takes an input sentence in the form of word embeddings and uses a series of recurrent neural networks to discover the basic set of nodes and subgraphs that comprise the AMR concepts and the set of predicate-argument relations among those concepts. With this configuration, the authors reached a Smatch f-score of 70.7% on the LDC2015E86 corpus.

Lyu and Titov (2018) introduced a neural parser that treats alignments as latent variables within a joint probabilistic model of concepts, relations, and alignments. Their parser requires five different bidirectional LSTMs aiming to identify concepts, relations, the root, and alignments. The parser reached a Smatch f-score of 73.7% and 74.4% on the LDC2015E86 and LDC2016E25 corpora, respectively.

[6] To encode graphs as linear sequences.

3.4 Parsers for other languages

These previous works focused on approaches for English corpora due to the lack of large annotated corpora for non-English language. Thus, a few studies attacked this gap.

Vanderwende et al (2015) produced a parser that may generate AMR graphs for sentences in French, German, Spanish, and Japanese, where AMR annotations were not available. For this end, they converted logical forms from an existing semantic analyzer (Vanderwende, 2015) into AMR graphs, using a set of rules. However, it was not possible to evaluate the systems, as there are no annotated corpora.

Damonte and Cohen (2018) proposed a method based on annotation projection, which involves exploiting annotations in a source language and a parallel corpus of the source language and a target language. Using English as the source language, the authors produced AMR graphs in Italian, Spanish, German, and Chinese target languages. Overall, the obtained results are still far from the parsers for English, reaching a Smatch f-score of 43%, 42%, 39%, and 35% for Italian, Spanish, German, and Chinese, respectively.

Wang et al (2018) adapted the parser of Wang et al (2015b) for the Chinese language and evaluated their approach on the annotated Chinese corpus (Li et al, 2016), achieving a Smatch f-score of 58.7%.

Anchiêta and Pardo (2018a) developed a rule-based AMR parser for Portuguese. The authors proposed generic rules to convert a dependency tree with semantic role information into AMR graph. They reached a Smatch f-score of 53.5% on the annotated Portuguese corpus (Anchiêta and Pardo, 2018b).

3.5 Summary

We organized the aforementioned works by year of publication, as shown in Table 3. The 2013N and 2014N corpora refer to the newswire section of the LDC2013E117 and LDC2014T12 corpora, respectively. In addition, we highlighted the main results for each corpus and, since the LDC2016E25 and LCD2017T10 corpora are equal[7], we emphasized only the LDC2016E25 corpus, as a parser of Lyu and Titov (2018) obtained better results in this corpus. It is important to see that the transition-system based approach followed by the tree approach obtained better results for small corpora, while deep learning performed better for large corpora. For instance, the parsers of Zhou et al (2016) and Guo and Lu (2018), which adopt a transition-system based approach, achieved an f-score of 71% and 74% on the LDC2013N and LDC2014N corpora, respectively; the parsers of Goodman et al (2016) and Wang et al (2015a), which use tree-based approaches, reached an f-score of 70% on the LDC2013N corpus; the parsers of van Noord and Bos (2017) and Lyu and Titov (2018), which use deep learning approaches, achieve f-scores of 68.5%

[7] The corpora have the same sentences, hence the same annotation, as introduced in subsection 3.1

and 73.4% on the LDC2015 corpus, 71% and 74.4% on the LDC2016 corpus, respectively.

Table 3 Timeline of AMR English parsers and results

Parser	Year	Approach	Corpus - F1 (%)					
			2013N	2014N	2014	2015	2016	2017
Flanigan et al (2014)	2014	Graph	58	-	-	-	-	-
Werling et al (2015)	2015	Graph	62.3	62.2	-	-	-	-
Wang et al (2015b)	2015	Tree	63	-	-	-	-	-
Wang et al (2015a)	2015	Tree	70	70	66	-	-	-
Artzi et al (2015)	2015	CCG	-	66.3	-	-	-	-
Goodman et al (2016)	2016	Tree	70	-	-	-	-	-
Zhou et al (2016)	**2016**	**Transition**	**71**	71	66	-	-	-
Misra and Artzi (2016)	2016	CCG	-	66.1	-	-	-	-
Peng et al (2017)	2017	Deep Learning	-	-	-	52	-	-
Damonte et al (2017)	2017	Transition	-	-	-	64	-	-
Konstas et al (2017)	2017	Deep Learning	-	-	-	62.1	-	-
Foland and Martin (2017)	2017	Deep Learning	-	-	-	70.7	-	-
Wang and Xue (2017)	2017	Tree	-	-	68	68.1	-	-
Ballesteros and Al-Onaizan (2017)	2017	Transition	-	69	64	-	-	-
van Noord and Bos (2017)	2017	Deep Learning	-	-	-	68.5	71	-
Peng et al (2018)	2018	Transition	-	-	-	64	-	-
Vilares and Gómez-Rodríguez (2018)	2018	Transition	-	-	-	64	-	-
Lyu and Titov (2018)	**2018**	**Deep Learning**	-	-	-	**73.7**	**74.4**	-
Guo and Lu (2018)	**2018**	**Transition**	-	**74**	**68.3**	68.7	-	69.8

We also summarized the results of AMR parsers for other languages, as presented in Table 4. We did not insert results in the multilingual parser of Vanderwende et al (2015), as the authors did not evaluate their parser. Despite of Damonte and Cohen (2018) evaluated their parser for Italian, Spanish, German, and Chinese languages, they do not have gold-annotated corpora for these languages. Only the works of Anchiêta and Pardo (2018a) and Wang et al (2018) evaluated their parsers on manually annotated corpora for Portuguese and Chinese, respectively.

Table 4 Timeline of non-English parsers and results

Language	Parser	Year	Approach	Corpus - F1 (%)				
				Italian	Spanish	German	Chinese	Portuguese
Multilingual	Vanderwende et al (2015)	2015	Multilingual	-	-	-	-	-
Multilingual	Damonte and Cohen (2018)	2018	Multilingual	43	42	39	35	-
Portuguese	Anchiêta and Pardo (2018a)	2018	Rule	-	-	-	-	53.5
Chinese	Wang et al (2018)	2018	Tree	-	-	-	58.7	-

In the next section, we detail the adapted parsers for Portuguese and present our proposed improvements for the parser developed for this language.

4 AMR Parsers

The annotated corpus for Portuguese has 1,527 sentences aligned with *The Little Prince* version for English (Anchiêta and Pardo, 2018b). Thus, we adapted approaches based on transition-system and tree since they obtained better results for small corpora. Nevertheless, we also adapted an approach based on deep learning, aiming to investigate how a more robust parser deals with a

small corpus. Finally, we added rules to the parser proposed by Anchiêta and Pardo (2018a), trying to improve it.

For approaches based on transition-system and tree, we adapted the parsers of Damonte et al (2017) and Wang et al (2015b,a) (henceforth, we refer to them as `AMREager` and `CAMR`), respectively, as they are open source, need only minor modification for re-use with other languages, and have a good performance on small corpora. They require tokenization, lemmatization, Part-Of-Speech (POS) tagging, chunking parsing, dependency parsing, and Named Entity Recognition (NER), which for Portuguese are provided by Natural Language Toolkit (NLTK) (Loper and Bird, 2002) , PALAVRAS parser (Bick, 2000), NLPnet tagger (Fonseca and Rosa, 2013), LX-Parser (Silva et al, 2010), CoreNLP (Manning et al, 2014), and spaCy[8], respectively. These parsers also use several lexical resources, such as a list of countries, states and cities, negative words, and, finally, pre-trained word embeddings. For these lexical resources, we used their respective translations and pre-trained word embeddings for Portuguese (Hartmann et al, 2017).

For training these parsers, it is necessary to align the nodes of AMR graph with the tokens of the sentence. For that, the parsers use the JAMR aligner (Flanigan et al, 2014), which produces alignments as the one in Figure 6.

```
:: 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))
```

Fig. 6 Example of alignment between an AMR graph and tokens of the sentence "*Flowers have thorns just for spite!*".

The format of the alignment is a space separated list of spans with their graph fragment, where each node is specified by a descriptor: `0` for the root node, `0.0` for the first child of the root node, `0.1` for the second child of the root node and so forth. In this example, the span `0-1` (that is the token `Flowers`) is aligned with node `0.0` (that indicates the first child of the root node). This alignment was developed for English, which has slightly different tokenization in relation to Portuguese. For example, for Portuguese, some hyphenated words as "*perturbou-me, achou-me, ouvi-lo*" (translated for "*disturbed me, found me, hear him*") should be separated by the hyphen, whereas other words as "*guarda-chuva, mal-humorado, recém-casados*" (translated for "*umbrella, grumpy, newly married*") should not be separated. The JAMR aligner does not align these cases. Thus, in order to overcome this concern, we manually reviewed the produced alignments. Furthermore, the JAMR aligner does not align reentrancies, that is, when a node takes part in multiple se-

[8] `https://spacy.io/`

mantic relations in the graph. To solve this alignment issue, we duplicated nodes that have a reentrant relation, as depicted in Figure 7. One may see on the left that the aligner does not align the concept `himself` to the node `he`. So, we duplicated the node `he` (on the right) to handle this problem. The post-processing will restore reentrant relations.

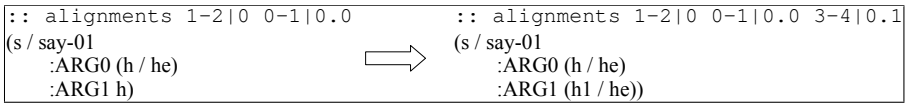


Fig. 7 Duplicating the node `he` (on the right) in reentrant relation for the sentence "*He says to himself.*".

Besides the improvement in the alignment, the `CAMR` parser uses the value of the Smatch metric to choose the best model in the training phase. For that, its algorithm learns the features of the training set and evaluates them on the development set. Then, it adopts the Smatch metric to evaluate the development set and to choose the best model. However, this metric overlooks several parsing problems and assigns high scores for sentences with different meanings. For example, the sentences "*I have two houses*" and "*She bought three cars*" may be represented as shown in Figure 8, and using the Smatch metric to compare their AMRs, it returns an f-score of 0.29, which we consider too high for so different sentences. In this way, the `CAMR` parser may choose a worse model because of the overvaluing of the Smatch metric.

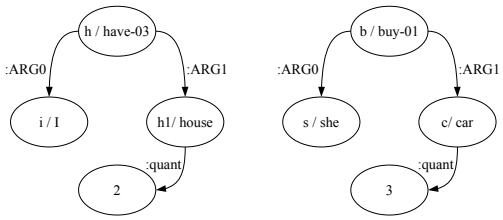


Fig. 8 In the left, AMR graph for the sentence "*I have two houses*" and, in the right, AMR graph for the sentence "*She has three cars.*"

For helping that parser to choose a better model, we replaced the Smatch metric by the SEMA metric (Anchiêta et al, 2019). SEMA metric solves this problem taking into account the dependency of the nodes in the graph. For that example, the SEMA metric returns an f-score of 0.0, being fairer than the Smatch metric.

For the deep learning approach, we adapted the parser of van Noord and Bos (2017) (henceforth, we refer to it as `NeuralAMR`), using the OpenNMT system (Klein et al, 2017). This parser adopts a *seq2seq* model with bidirectional encoding and general attention (Luong et al, 2015) with detailed settings shown in Table 5.

Table 5 Parameter settings of the seq2seq model

Parameter	Value	Parameter	Value
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

Besides these settings, the model learns embeddings during the training phase, as the authors used a large corpus. More than that, they introduced an approach called "super characters" that is a combination of word and character level input. For instance, the authors transformed AMR relations (e.g. `:ARG0`, `:ARG1`) as atomic instead of a set of characters (e.g. *A*, *R*, *G*, *0* for `:ARG0`). And, for sentences, they incorporated syntactic information into the characters of the sentence, as depicted in Figure 9. Their parser transforms the words of the input sentence, for example, "are", into characters (*a*, *r*, *e*) with atomic POS-tags (*VBP*). For AMR annotation, they transform the AMR concepts, for instance, "that", into characters (*t*, *h*, *a*, *t*), and AMR relations into atomic information.

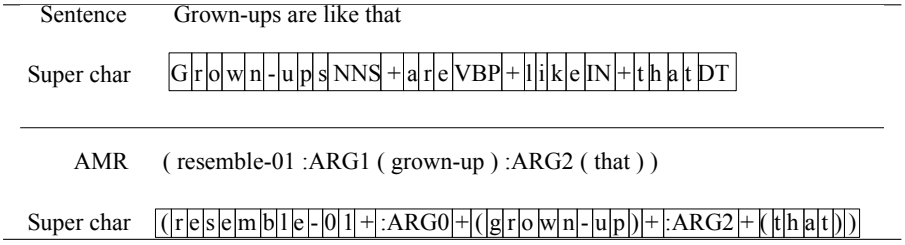


Fig. 9 Input for the sentence "*Grown-ups are like that*" with POS-tags and input for the AMR `(resemble-01 :ARG1 (grown-up) :ARG2 (that)))`. The + symbol represents spaces.

We may see that, to adopt a *seq2seq* model, it is required to linearize the AMR graph. Another change in the AMR structure is the removal of the variables of each concept, since the model does not need to learn them. We recover this information in the post-processing stage. Moreover, this approach does not need alignment between the word tokens in the sentence and the concepts in the AMR graph.

As our corpus is much smaller than that used by the authors, we added more information to the super character structure, such as lemma, dependency relation, and named entity. Besides that, we used pre-trained embeddings (Hartmann et al, 2017), aiming to improve the model.

In addition to these two adapted parsers for Portuguese, we improved an AMR parser developed for that language (Anchiêta and Pardo, 2018a) (henceforth, we refer to it as `RBAMR`). This parser solves the problem of generating an

AMR structure applying a set of generic rules on a pre-processed input text with POS and NER tagging and Semantic Role Labeling (SRL) information, transforming it into an AMR graph. The authors designed six rules that produce the following AMR relations: `named entity`, `:mod`, `:manner`, `:degree`, `:polarity`, and `:time`. Although these relations are the most frequent in the corpus, the authors did not treat two essential phenomena, which also appear with high frequency in the corpus: the verb to 'be' and conjunctions. Thus, we extended the parser to deal with these phenomena, with rules that we detail below.

- **The verb to 'be'.** This rule creates a `:domain` relation in sentences such as *... **be** <noun>* and *... **be** <adjective>* when there is no frame for that noun or adjective. Figure 10 shows an example of this case.
- **Conjunctions.** We treated two conjunction types: contrastive and additive. In the first, we produce the concept `contrast-01` for conjunctions, such as *but*, *while*, *whereas*, and *however*. In the second, we create the concepts `and` or `or`, as depicted in Figure 11.

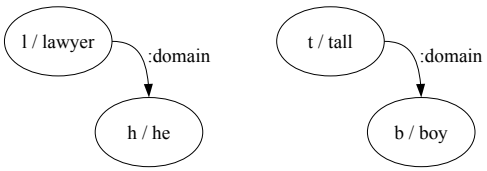


Fig. 10 Rule of the verb to 'be' for the sentences: "*He is a lawyer*" (left) and "*The boy is tall*" (right).

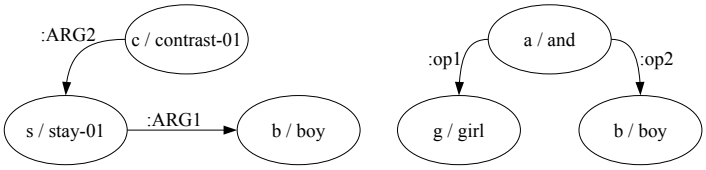


Fig. 11 Rule for conjunctions: concepts `contrast-01` and `and`, respectively, for the sentence: "*But the boy stayed*" (left) and "*The girl and boy*" (right).

In addition to these two rules, we applied a pruning method in the post-processing stage to increase the quality of the AMRs, since the parser does not keep track of what has already produced, generating redundant nodes. Hence, we removed all recurring nodes with the same parent.

4.1 Post-processing

The `CAMR` and `AMREager` parsers require the alignment among word tokens in the sentence and nodes in the graph. For that, they use the JAMR aligner. However, the aligner does not align reentrant relation. Therefore, in the pre-processing step, we duplicated nodes in reentrant relations to overcome this limitation of the aligner. Then, in the post-processing, we restore the reentrant relation. Figure 12 shows an example of nodes with reentrant relations restored.

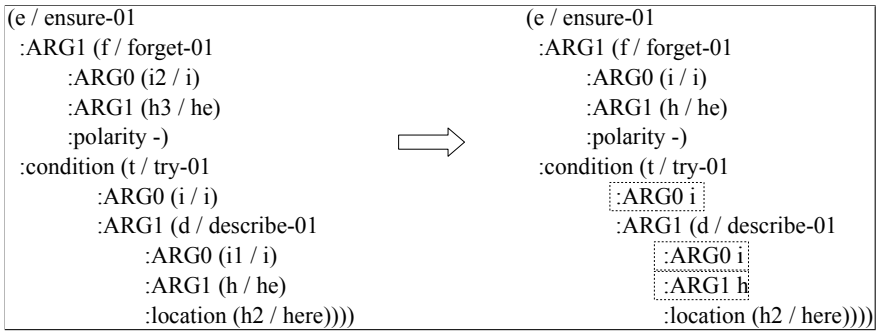


Fig. 12 An example of co-referring nodes restored. On the left, an example of duplicated nodes; on the right, the restored nodes `i` and `he`.

In the `RBAMR` parser, we applied a pruning method for removing redundant nodes of the same parent. In Figure 13, we show an example of the method.

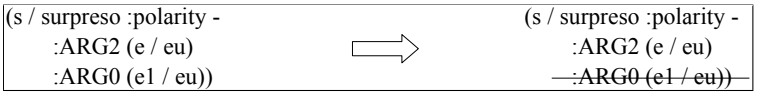


Fig. 13 An example of pruned nodes. On the left, an example of redundant nodes; on the right, the pruned node `I`.

In what follows, we evaluate these semantic parsers and present an error analysis on the annotated corpus for Portuguese.

5 Evaluation and Results

We conducted our experiments on the annotated corpus for Portuguese (Anchiêta and Pardo, 2018b). The corpus is aligned with the English version, keeping the original training/dev/test division proposed for that language[9]:

[9] `https://amr.isi.edu/download.html`

1,274, 145, and 143 sentences for training, development, and testing, respectively. Moreover, we split the corpus by sentence length since the longer the sentence is, the more difficult the semantic parsing is. Longer sentences are more difficult to preprocess and errors that occur in this stage propagate to the following steps. Besides, it is possible to get the weakness and strengths of the AMR parsers. Therefore, we calculated the average sentence length of the corpus and obtained the value of 10.46 tokens per sentence, wherein the test set, 80 sentences are shorter and 63 are longer than average.

For training the adapted parsers, we pre-processed each sentence using several tools. For the `CAMR` and `AMREager` parsers, we used NLTK, PALAVRAS parser, NLPnet tagger, LX-Parser, CoreNLP, and spaCy to get tokenization, lemmatization, POS tags, constituent trees, dependency relation, and NER tags as features, respectively. For the `NeuralAMR` parser, we used NLTK, PALAVRAS parser, NLPnet tagger, CoreNLP, and spaCy to obtain tokenization, lemmaization, POS tags, dependency relation, and NER tags as features, respectively. At last, for the `RBAMR` parser, we used PALAVRAS parser to get POS and NER tags, and Brazilis SLR (Hartmann et al, 2016) to obtain SLR information.

For evaluating these parsers, we adopted the Smatch and SEMA metrics. Figures 14 and 15 show f-score results for shorter and longer sentences for each parser and metric.

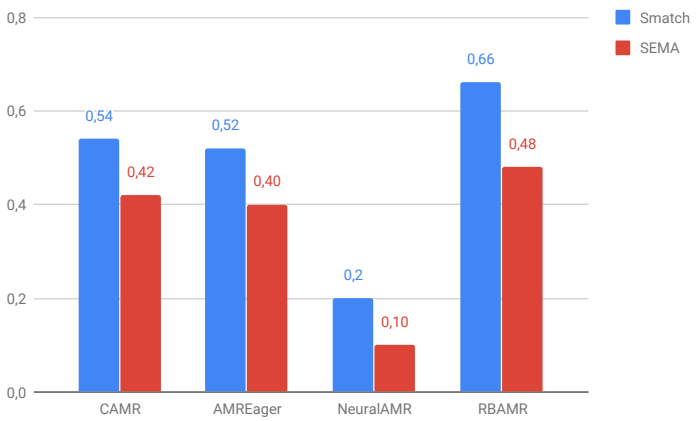


Fig. 14 Results for shorter sentences for Portuguese AMR parsers

In Figures 14 and 15, one may see that the `RBAMR` obtained the best values on both metrics and sentences length. The parser achieved 0.66 and 0.48 of f-score for shorter sentences on Smatch and SEMA metrics, respectively, and 0.49 and 0.28 of f-score for longer sentences for the same metrics. The `CAMR` and `AMREager` parsers had a similar result in both metrics and sentence length,

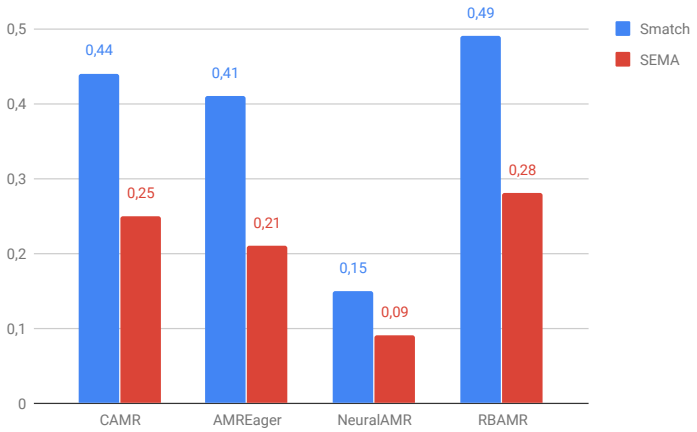


Fig. 15 Results for longer sentences for Portuguese AMR parsers

with the first being slightly higher than the second. Finally, the `NeuralAMR` parser did not achieve good results due to size of the corpus.

In Table 6, we present an ablation study to show the contributions proposed for each parser, giving f-score values of the Smatch and SEMA metrics. From this table, we may see that the rules plus the pruning method improved the previous `RBAMR` parser in 0.05 and 0.03 on Smatch metric for shorter and longer sentences, respectively. The improvements in the alignment also outperformed the original `CAMR` and `AMREager` parsers. Furthermore, adopting the SEMA metric, in the training phase of the `CAMR`, parser also improved the results.

Table 6 Contributions for the AMR parsers

Parser	Shorter Sentences		Longer Sentences	
	Smatch	Sema	Smatch	Sema
`RBAMR`	0.61	0.43	0.46	0.25
`RBAMR` + rules	0.62	0.44	0.48	0.27
`RBAMR` + pruning	0.64	0.46	0.47	0.26
`RBAMR` + rules + pruning	**0.66**	**0.48**	**0.49**	**0.28**
`CAMR`	0.51	0.40	0.40	0.22
`CAMR` + alignment	0.53	0.41	0.42	0.24
`CAMR` + SEMA	0.52	0.41	0.41	0.23
`CAMR` + alignment + SEMA	**0.54**	**0.42**	**0.44**	**0.25**
`AMREager`	0.50	0.39	0.39	0.19
`AMREager` + alignment	**0.52**	**0.40**	**0.41**	**0.21**

To show a comparison between English and Portuguese AMR parsers, we trained the original English AMR parsers (without adaptations) on the English version of the *Little Prince* corpus and compared with the best-adapted parsers trained on the Portuguese corpus (see Figure 16). From this figure, in all scenarios the English parsing models were better than Portuguese. On average,

the English parsers reached 0.46 and 0.32 of f-score for Smatch and SEMA metrics, respectively, while, for Portuguese, they achieved 0.37 and 0.24 for the same measure and metrics. The `English CAMR` parser obtained the best results in both metrics, achieving 0.57 and 0.39 of f-score, overcoming in 0.08 and 0.05 the `Portuguese CAMR` parser, which was the best model adapted for Portuguese.

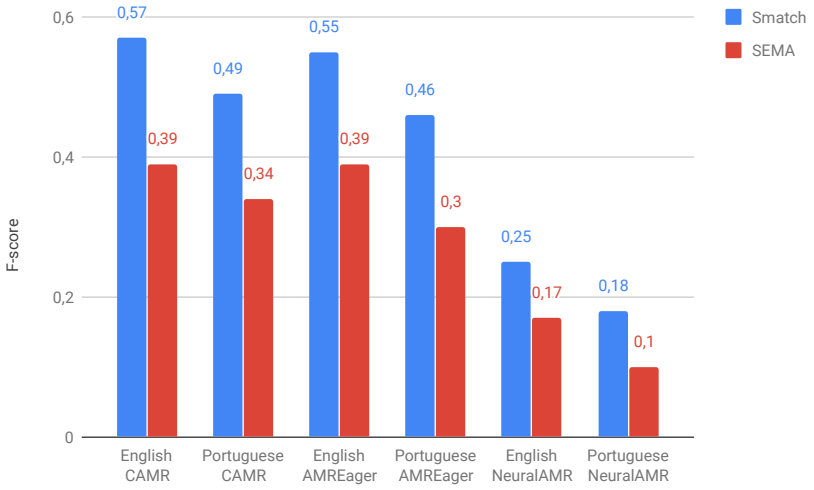


Fig. 16 Comparison between English and Portuguese AMR Parsing on the test set.

We also compared the original `CAMR` parsers against its modified version, i.e., `CAMR` + alignment + SEMA, and the original `AMREager` parser against `AMREager` + alignment. In Figure 17, we present the results. We may see that our adaptations improved AMR parsing results, outperforming the original version of the parsers in both Smatch and SEMA metrics by 0.02 points. It is interesting that a very simple modification may improve the results.

5.1 Error Analysis

We may see that AMR parsing is easier in shorter sentences than longer sentences, since the last is more prone to errors in the pre-processing. Moreover, the number of tools used in the pre-processing contribute to worse results. For example, for the `CAMR` and `AMREager` parsers, we used six tools to extract the same features from a sentence than CoreNLP (Manning et al, 2014) toolkit does for English. Moreover, these six tools have lower accuracy than CoreNLP.

Another important factor for a low f-score of `CAMR` and `AMREager` parsers are wrong alignments among the word tokens in the sentence and the concepts in the AMR graph. Even though we reviewed the alignments, some phenomena

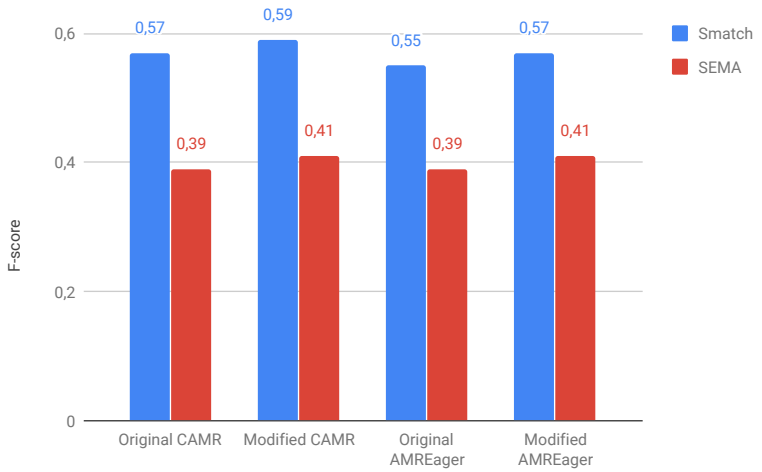


Fig. 17 Comparison between original `CAMR` and `AMREager` parsers against our modifications.

are difficult to solve, as hidden subject, that is, syntactic subjects that do not appear in the sentence (they are implicit), but are annotated to keep the similarity with English annotation (Anchiêta and Pardo, 2018b).

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

Fig. 18 Alignment between an AMR graph and word tokens for the sentence "*Preciso é de um carneiro.*" (What I need is a sheep.)

For instance, in the sentence "*Preciso é de um carneiro.*", the subject '*eu*' (the pronoun 'I' in English) does not appear in the sentence, but it was annotated, as depicted in Figure 18. Hence, this token does not have an alignment.

Moreover, observing the generated AMR graphs by the `CAMR` parser, it produces a `:null_edge` relation whenever it does not identify a proper relation. This case occurred 95 times, which represent 20% of the total number of relations, and, for that, if the parser changes the `:null_edge` relation to `:ARG0`, for instance, the result would improve 0.05 in f-score. The parser also produces a `null_tag` concept when it does not identify a concept in the sentence. This case appears only two times in the results, and it is harder to be solved. Figure 19 presents an example of these relations and concepts produced by the `CAMR` parser. We believe that these phenomena arise because of the small size of the training set, since they also occur in the `English CAMR` parser.

The `AMREager` parser generates an `emptygraph` concept when it does not identify concepts and relations in a sentence. This occurs six times in the

```
(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))
```

Fig. 19 `:null_edge` relation and `null_tag` concept produced by the `CAMR` parser for the sentence "*Quando é meio dia nos Estados Unidos, o sol, todo mundo sabe, está se deitando na França.*" (When it is noon in the United States the sun is setting over France.)

results, and it is responsible for the low result in concept and SRL identification compared to the `CAMR` parser, as shown in Figure 20. For instance, for the sentence "*Um dia eu vi o sol se pôr quarenta e três vezes!*" (One day, I saw the sunset forty-three times!), the parser generates only an `emptygraph` concept. We also believe that these cases appear due to the small size of the corpus.

The `NeuralAMR` parser, although it does not produce `:null_edge` relation and `null_tag` and `emptygraph` concepts, it did not performed well either for the Portuguese or the English version, because of the small size of the corpus.

The `RBAMR` parser got better results, as depicted in Figures 14 and 15. As this is a rule-based parser, it does not require the alignment among word tokes and concepts. Furthermore, the parser uses only two tools to get POS and NER tags and SRL information, avoiding more pre-processing errors.

In addition to the above errors, we also analyzed the errors in nodes (concepts) and edges (relations) identification to gain more insights. Nonetheless, an AMR graph has several possible relations (over 100), and analyzing each relation is a laborious task. So, we break down the AMR parsing task in two sub-tasks: concept and SRL identification to facilitate our analysis. For that, we used the evaluation tool from Damonte et al (2017) for reporting the performance for these components. Figure 20 presents the results for the concept and SRL identification. From this figure, one may see that the `RBAMR` parser got better results both in tasks and metrics. We believe that this is due to the parser being aligner-free and using few tools in the pre-processing stage, helping to reach better results than the adapted parsers. Although the `CARM` parser did not achieve better results than `RBAMR` parser in this corpus, we believe that in a larger corpus the `CAMR` parser may overcome it.

6 Conclusion

In this paper, we presented an adaptation of some AMR parsing models from English to Portuguese and improvements of a previous AMR parser designed for Portuguese. We evaluated these models on a manually annotated corpus for Portuguese. Besides the adaptation, we improved two AMR parsing models, duplicating nodes in the reentrant relations, restoring them in post-processing, and changing the metric of choosing the best model in the training phase. We

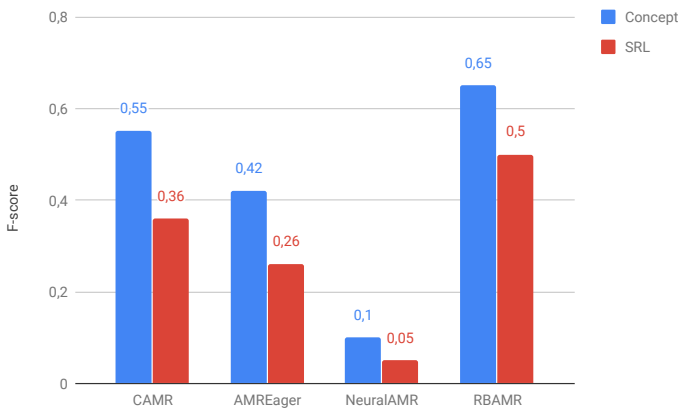


Fig. 20 Fine-grained results for the concept and SRL identification

also improved the previous AMR parser designed to Portuguese, adding rules into it and applying a pruning method, in post-processing, on their duplicate nodes. More than that, we presented detailed results showing the contributions of our improvements. Finally, we performed a detailed error analysis, giving some insights for future work in AMR parsing for Portuguese.

Acknowledgements The authors are grateful to USP Research Office (PRP 668) and IFPI for supporting this work.

References

Abend O, Rappoport A (2013) Universal conceptual cognitive annotation (ucca). In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 228–238

Anchiêta RT, Pardo TAS (2018a) A rule-based amr parser for portuguese. In: Simari GR, Fermé E, Gutiérrez Segura F, Rodríguez Melquiades JA (eds) Advances in Artificial Intelligence - IBERAMIA 2018, pp 341–353

Anchiêta RT, Pardo TAS (2018b) Towards amr-br: A sembank for brazilian portuguese. In: Proceedings of the 11th edition of the Language Resources and Evaluation Conference, pp 974–979

Anchiêta RT, Cabezudo MAS, Pardo TAS (2019) SEMA: an extended semantic evaluation metric for amr. In: (To appear) Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing

Artzi Y, Lee K, Zettlemoyer L (2015) Broad-coverage ccg semantic parsing with amr. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp 1699–1710

Ballesteros M, Al-Onaizan Y (2017) Amr parsing using stack-lstms. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 1269–1275

Banarescu L, Bonial C, Cai S, Georgescu M, Griffitt K, Hermjakob U, Knight K, Palmer M, Schneider N (2013) Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pp 178–186

Bick E (2000) The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus Universitetsforlag

Bos J (2016) Expressive power of abstract meaning representations. Computational Linguistics pp 527–535

Burns GA, Hermjakob U, Ambite JL (2016) Abstract meaning representations as linked data. In: International Semantic Web Conference, pp 12–20

Cai S, Knight K (2013) Smatch: an evaluation metric for semantic feature structures. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp 748–752

Covington MA (2001) A fundamental algorithm for dependency parsing. In: Proceedings of the 39th annual ACM southeast conference, pp 95–102

Damonte M, Cohen SB (2018) Cross-lingual abstract meaning representation parsing. In: Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies, pp 1146–1155

Damonte M, Cohen SB, Satta G (2017) An incremental parser for abstract meaning representation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp 536–546

Flanigan J, Thomson S, Carbonell JG, Dyer C, Smith NA (2014) A discriminative graph-based parser for the abstract meaning representation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp 1426–1436

Foland W, Martin JH (2017) Abstract meaning representation parsing using lstm recurrent neural networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 463–472

Fonseca ER, Rosa JLG (2013) Mac-morpho revisited: Towards robust part-of-speech tagging. In: Proceedings of the 9th Brazilian symposium in information and human language technology, pp 98–107

Goodman J, Vlachos A, Naradowsky J (2016) Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 1–11

Guo Z, Lu W (2018) Better transition-based amr parsing with a refined search space. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp 1712–1722

Hardy H, Vlachos A (2018) Guided neural language generation for abstractive summarization using abstract meaning representation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp 768–773

Hartmann N, Fonseca E, Shulby C, Treviso M, Silva J, Aluísio S (2017) Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology, pp 122–131

Hartmann NS, Duran MS, Aluísio SM (2016) Automatic semantic role labeling on non-revised syntactic trees of journalistic texts. In: International Conference on Computational Processing of the Portuguese Language, pp 202–212

Issa F, Damonte M, Cohen SB, Yan X, Chang Y (2018) Abstract meaning representation for paraphrase detection. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), vol 1, pp 442–452

Jurafsky D, Martin J (2009) Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall

Kingsbury P, Palmer M (2002) From treebank to propbank. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation, pp 1989–1993

Klein G, Kim Y, Deng Y, Senellart J, Rush A (2017) Opennmt: Open-source toolkit for neural machine translation. Proceedings of ACL 2017, System Demonstrations pp 67–72

Konstas I, Iyer S, Yatskar M, Choi Y, Zettlemoyer L (2017) Neural amr: Sequence-to-sequence models for parsing and generation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 146–157

Kuhlmann M, Jonsson P (2015) Parsing to noncrossing dependency graphs. Transactions of the Association for Computational Linguistics 3:559–570

Lehmann F (1992) Semantic networks in artificial intelligence. Elsevier Science Inc.

Li B, Wen Y, Weiguang Q, Bu L, Xue N (2016) Annotating the little prince with chinese amrs. In: Proceedings of the 10th Linguistic Annotation Workshop, pp 7–15

Liu F, Flanigan J, Thomson S, Sadeh N, Smith NA (2015) Toward abstractive summarization using semantic representations. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1077–1086

Loper E, Bird S (2002) Nltk: The natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1, pp 63–70

Luong T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp 1412–1421

Lyu C, Titov I (2018) Amr parsing as graph prediction with latent alignment. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 397–407

Manning CD, Surdeanu M, Bauer J, Finkel J, Bethard SJ, McClosky D (2014) The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations, pp 55–60, URL http://www.aclweb.org/anthology/P/P14/P14-5010

Matthiessen C, Bateman JA (1991) Text generation and systemic-functional linguistics: experiences from English and Japanese. Pinter Publishers

Migueles-Abraira N, Agerri R, de Ilarraza AD (2018) Annotating abstract meaning representations for spanish. In: Proceedings of the 11th International Conference on Language Resources and Evaluation, pp 3074–3078

Misra DK, Artzi Y (2016) Neural shift-reduce ccg semantic parsing. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp 1775–1786

Mitra A, Baral C (2016) Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In: Proceedings of the 30th Conference on Artificial Intelligence, pp 2779–2785

Nivre J (2004) Incrementality in deterministic dependency parsing. In: Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together, pp 50–57

van Noord R, Bos J (2017) Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. Computational Linguistics in the Netherlands Journal 7:93–108

Palmer M, Gildea D, Kingsbury P (2005) The proposition bank: An annotated corpus of semantic roles. Computational linguistics 31(1):71–106

Pan X, Cassidy T, Hermjakob U, Ji H, Knight K (2015) Unsupervised entity linking with abstract meaning representation. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1130–1139

Peng X, Wang C, Gildea D, Xue N (2017) Addressing the data sparsity issue in neural amr parsing. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, vol 1, pp 366–375

Peng X, Gildea D, Satta G (2018) Amr parsing with cache transition systems. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp 4897–4904

Pourdamghani N, Knight K, Hermjakob U (2016) Generating english from abstract meaning representations. In: International Conference on Natural Language Generation, pp 21–25

Silva J, Branco A, Castro S, Reis R (2010) Out-of-the-box robust parsing of portuguese. In: Pardo TAS, Branco KA António, Vieira R, de Lima VLS (eds) International Conference on Computational Processing of the Portuguese Language, Springer, pp 75–85

Song L, Peng X, Zhang Y, Wang Z, Gildea D (2017) Amr-to-text generation with synchronous node replacement grammar. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp 7–13

Song L, Zhang Y, Wang Z, Gildea D (2018) A graph-to-sequence model for amr-to-text generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1616–1626

Song L, Gildea D, Zhang Y, Wang Z, Su J (2019) Semantic neural machine translation using AMR. Transactions of the Association for Computational Linguistics 7:19–31, DOI 10.1162/tacl_a_00252, URL https://www.aclweb.org/anthology/Q19-1002

Steedman M (1996) Surface Structure and Interpretation. Linguistic inquiry monographs, MIT Press

Steedman M (2001) The Syntactic Process. A Bradford book, MIT Press

Uchida H, Zhu M, Della Senta T (1996) Unl: Universal networking language–an electronic language for communication, understanding, and collaboration. Tokyo: UNU/IAS/UNL Center

Vanderwende L (2015) Nlpwin–an introduction. Tech. rep., Microsoft Research tech report no. MSR-TR-2015-23

Vanderwende L, Menezes A, Quirk C (2015) An amr parser for english, french, german, spanish and japanese and a new amr-annotated corpus. In: Proceedings of the 2015 Meeting of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies, pp 26–30

Vilares D, Gómez-Rodríguez C (2018) A transition-based algorithm for unrestricted amr parsing. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), vol 2, pp 142–149

Vinyals O, Kaiser Ł, Koo T, Petrov S, Sutskever I, Hinton G (2015) Grammar as a foreign language. In: Advances in neural information processing systems, pp 2773–2781

Wang C, Xue N (2017) Getting the most out of amr parsing. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 1257–1268

Wang C, Xue N, Pradhan S (2015a) Boosting transition-based amr parsing with refined actions and auxiliary analyzers. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp 857–862

Wang C, Xue N, Pradhan S, Pradhan S (2015b) A transition-based algorithm for amr parsing. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 366–375

Wang C, Li B, Xue N (2018) Transition-based chinese amr parsing. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,

Volume 2 (Short Papers), vol 2, pp 247–252
Werling K, Angeli G, Manning CD (2015) Robust subgraph generation improves abstract meaning representation parsing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol 1, pp 982–991
Zhou J, Xu F, Uszkoreit H, Qu W, Li R, Gu Y (2016) Amr parsing with an incremental joint model. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp 680–689

CHAPTER

8

SEMANTICALLY INSPIRED AMR ALIGNMENT FOR THE PORTUGUESE LANGUAGE

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo. 2020. “Semantically Inspired AMR Alignment for the Portuguese Language”, accepted in the Proceedings of the 9th Brazilian Conference on Intelligent Systems (BRACIS).

Contribution statement

R.T. Anchiêta conceived and developed the research and contributed in writing the manuscript. T.A.S. Pardo helped in conceiving the research and writing the manuscript and supervised the project.

Semantically Inspired AMR Alignment for the Portuguese Language

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo

Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences (ICMC)
University of São Paulo (USP), São Carlos, SP, Brazil
rta@usp.br, taspardo@icmc.usp.br

Abstract. Abstract Meaning Representation (AMR) is a graph-based semantic formalism where the nodes are concepts and edges are relations among them. Most of AMR parsing methods require alignment between the nodes of the graph and the words of the sentence. However, this alignment is not provided by manual annotations and available automatic aligners focus only on the English language, not performing well for other languages. Aiming to fulfill this gap, we developed an alignment method for the Portuguese language based on a more semantically matched word-concept pair. We performed both intrinsic and extrinsic evaluations and showed that our alignment approach outperforms the alignment strategies developed for English, improving AMR parsers, and achieving competitive results with a parser designed for the Portuguese language.

Keywords: Natural Language Processing · Abstract Meaning Representation · Alignment.

1 Introduction

According to Banarescu et al. [4], Abstract Meaning Representation (AMR) is a semantic meaning representation, which may be encoded as a rooted Direct Acyclic Graph (DAG) where the nodes are concepts and the edges are relations among them. In addition to a graph notation, it is common to see AMR as a serialized graph in PENMAN notation [20]. This representation explicitly details semantics information, as depicted in Figure 1. In this figure, the `live-01` node is the root of the graph and `city` node introduces a named entity. Moreover, `:ARGx` relations are predicates from the PropBank lexicon [14], which encode semantic information according to each PropBank sense.

To parse a text into an AMR graph, most of the AMR parsers require alignment between the word (tokens) of the sentence and the nodes of the corresponding graph (see, for instance, [10, 27, 30, 8]). However, this anchoring is not provided by manual annotations, i.e., annotators are not responsible to produce that alignment between words of a sentence and nodes of the corresponding graph. In addition, the available automatic aligners focus only on the English language

```
(l / live-01
   :ARG0 (i / i)
   :location (c / city
      :name (n / name
         :op1 "Barcelona")))
```

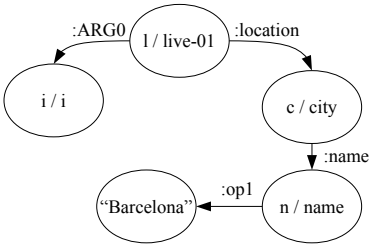


Fig. 1. PENMAN notation (left) and graph notation (right) for the sentence "I live in Barcelona".

[23, 10, 18] and they do not perform well for other languages. For Portuguese, for instance, the sentence "*Não era surpresa para mim*" (It was no surprise to me), the `JAMR` aligner [10] produces alignment only between the token *surpresa* (surprise) and the node `surpresa`, as shown in Figure 2.

```
# ::snt Não era surpresa para mim
# ::alignments 2-3|0

(s / surpresa
   :polarity -
   :domain (e / eu))
```

Fig. 2. Alignment produced by the JAMR aligner for the sentence "*Não era surpresa para mim*" (It was no surprise to me).

The `JAMR` aligned only the span `2-3`, which is the token *surpresa* (surprise), with the node `0`, which is the root of the graph. The nodes `-` and `eu` were not aligned. This wrong alignment occurs because of the `JAMR` aligner adopts a string-match strategy that is focused on the English language. Thus, these issues contribute for decreasing the performance of AMR parsers. As a result, recent AMR parsing methods have focused on alignment-free approaches [19, 28, 29]. However, they require a large annotated corpus, which is available only for English.

In this context, aiming to bridge this lack of resources and tools for other languages, we propose an AMR aligner for Portuguese that focuses on a more semantically matched word-concept pair. For that, we use pretrained word embeddings and the Word Mover's Distance (WMD) function [15] to match span tokens in the sentence with nodes in the graph. Word embeddings capture some semantics information about a corpus, and WMD measures the dissimilarity between two documents even if they have no words in common. With this, it is possible to produce semantically inspired matches instead of only string-match.

To evaluate our approach, we carry out both intrinsic and extrinsic experiments on an annotated corpus from Portuguese. Our aligner produced better

alignments than alignment strategies proposed for English and improved AMR parsing for Portuguese, reaching competitive results with an AMR parser designed for that language.

The remaining of this paper is organized as follows. Section 2 presents the fundamentals of the AMR formalism. In Section 3, we briefly introduce the related work. Section 4 describes our proposed aligner. In Section 5, we conduct some experiments and evaluations, and show our results. Finally, Section 6 concludes the paper indicating future research.

2 AMR Background

Abstract Meaning Representation (AMR) is a relatively recent semantic formalism, whose objective is to capture the meaning of a sentence through a rooted Direct Acyclic Graph (DAG) [4], as presented in Figure 1. In addition to graph notation, AMR may be represented as a serialized graph in PENMAN notation [20] and a sequence of triples, as depicted in Figure 3.

```
(l / live-01
   :ARG0 (i / i)
   :location (c / city
      :name (n / name
         :op1 "Barcelona")))
```

```
instance (a, live-01 ^
instance (b, i) ^
instance (c, city) ^
instance (d, name)^
ARG0 (a, b) ^
location (a, c) ^
name (c, d) ^
op1 (d, "Barcelona")
```

Fig. 3. PENMAN notation (left) and sequence of triples (right) for the sentence "I live in Barcelona"

From this figure, `live-01` (`-01` indicates the sense of the verb), `i`, `city`, and `name` are concepts, `l`, `i`, `c`, and `n` are variables to index a concept, and "`Barcelona`" is a constant, as it gets no variable. Moreover, `live-01` and `i` are concrete concepts, since they are in the sentence, whereas `city` and `name` are abstract concepts. The relations are: `:ARG0`, which is adopted from the PropBank resource [14], meaning that the `i` concept is the agent in the sentence, `:location`, `:name`, and `:op1` are specific AMR-relations that encoded some semantic information.

The AMR representation is a graph because it allows reentrancy relations, which are multiple incoming in the graph. These types of relations are useful to avoid the creation of duplicate nodes. Figure 4 presents an example of a reentrancy relation for the sentence "The dog wants to eat the bone". In this figure, the reentrancy relation is represented by the edge between the `eat-01` and `dog` nodes.

To evaluate AMR structures, traditionally one uses the Smatch metric [7]. It assesses both inter-annotator agreement and automatic parsing accuracy, computing the degree of overlap between two AMR structures, calculating precision,

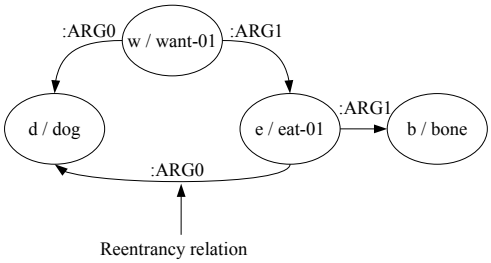


Fig. 4. An example of reentrancy relation for the sentence "The dog wants to eat the bone".

recall, and f-score over AMR triples. For that, Smatch tries to find the one-to-one node mapping between two AMR structures. For computing precision, recall, and f-score, it follows Equations 1, 2, and 3, respectively.

$$P = \frac{M}{C} \qquad (1) \qquad\qquad R = \frac{M}{T} \qquad (2) \qquad F1 = \frac{2 \times P \times R}{(P + R)} \qquad (3)$$

where M is the correct (according to a reference) number of triples, C is the produced number of predicted triples, and T is the total number of triples in AMR reference. For example, computing the AMR graph in the left against the AMR graph in the right of Figure 5, the Smatch metric produces the results as in Figure 6. It considers as correct the triples `instance (a, want-01)`, `instance (b, boy)`, `TOP (a, want-01)`, `ARG0 (a, b)`, and `ARG1 (a, c)`.

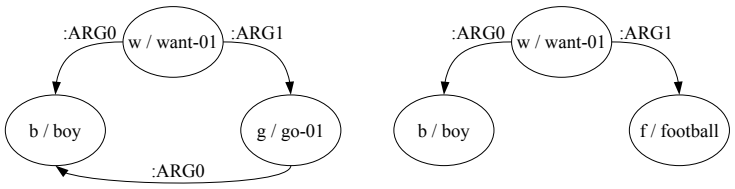


Fig. 5. Reference AMR graph (left) and Hypothesis AMR graph (right)

Anchiêta et al. [2] developed the SEMA metric that uses the same equations as Smatch. However, instead of finding the one-to-one node mapping, this metric takes into account the parent of the node under analysis. Furthermore, it removes the `TOP` triple used by the Smatch metric for computing precision, recall, and f-score. Considering the same example of Figure 5, SEMA generates the results according to Figure 7. The metric considers as correct the triples `instance (a, want-01)`, `instance (b, boy)`, `ARG0 (a, b)`, and `ARG1 (a, c)`.

Triples of the reference	Triples of the hypothesis	Smatch score
instance (a, want-01) ^ instance (b, boy) ^ instance (c, go-01) ^ TOP (a, want-01) ^ ARG0 (a, b) ^ ARG1 (a, c) ^ ARG0 (b, c)	instance (a, want-01) ^ instance (b, boy) ^ instance (c, football) ^ TOP (a, want-01) ^ ARG0 (a, b) ^ ARG1 (a, c)	P = 5 / 6 = 0.83 R = 5 / 7 = 0.71 F = 0.77

Fig. 6. Triples and results produced by the Smatch metric

Triples of the reference	Triples of the hypothesis	SEMA score
instance (a, want-01) ^ instance (b, boy) ^ instance (c, go-01) ^ ARG0 (a, b) ^ ARG1 (a, c) ^ ARG0 (b, c)	instance (a, want-01) ^ instance (b, boy) ^ instance (c, football) ^ ARG0 (a, b) ^ ARG1 (a, c)	P = 4 / 5 = 0.80 R = 4 / 6 = 0.66 F = 0.72

Fig. 7. Triples and results produced by the SEMA metric

For more details about the AMR formalism, we suggest consulting the original paper Banarescu et al. [4] and the guidelines[1].

3 Related Work

Flanigan et al. [10] developed the first AMR aligner, named `JAMR`. The authors created a rule-based aligner with fourteen heuristic rules to greedily align concepts in the nodes of the graph with tokens in the sentence. The alignment format is a space separated list of spans with their graph fragments, where each node is specified by a descriptor (e.g. Gorn (1965)[12]): `0` for the root node, `0.0` for the first child of the root node, `0.1` for the second child of the root node and so forth. For example, for the sentence, "The boy wants to go.", the `JAMR` generates alignments according to Figure 8. The `JAMR` aligned the spans `2-3`, `4-5`, and `1-2` (that refer to `wants`, `go`, and `boy`, respectively) with the nodes `0`, `0.1`, and `0.0` (that are the root of the graph, the second child of the root, and the first child of the root, respectively).

```
# ::snt The boy wants to go
# ::alignments 2-3|0 4-5|0.1 1-2|0.0

(w / want-01
   :ARG0 (b / boy)
   :ARG1 (g / go-01
      :ARG0 b))
```

Fig. 8. An example of the `JAMR` aligner for the sentence "The boy wants to go".

[1] `https://github.com/amrisi/amr-guidelines/blob/master/amr.md`

Pourdamghani et al. [23] adopted an unsupervised word alignment technique with machine learning. The authors followed a syntax-based Statistical Machine Translation (SMT) according to IBM word alignment model [6] to align linearized AMR graphs with English sentences. For the sentence "The boy wants to go", this approach produces an alignment as shown in Figure 9, where '∼ n' specifies a link to the nth English word. As we can see, the third token (`wants`) was aligned with the concept `want-01`, the second token (`boy`) with the concept `boy`, and the fifth token (`go`) with the concept `go-01`.

```
The boy wants to go

(w / want-01 ~ 3
   :ARG0 (b / boy ~ 2)
   :ARG1 (g / go-01 ~ 5
      :ARG0 b))
```

Fig. 9. Alignments produced for the sentence "The boy wants to go".

Liu et al. [18] extended and improved the `JAMR` aligner by adding semantic resources into the rules, such as Glove embeddings [22] and the Morphosemantic database [9]. Besides, they noted that the `JAMR` aligner requires that words have at least a common longest prefix of four characters, omitting the shorter cases (as the word `actions` that is not aligned with the concept `act-01`). Thus, their method improved the `JAMR` aligner in 4.6% f-score on the LDC2014T2 corpus. The authors also showed that their aligner improved the `JAMR` [10] and `CAMR` [27] parsers.

4 Our Aligner

In order to properly adapt AMR parsers from English to Portuguese, we developed an alignment strategy based on document similarity for this language. Our method produces alignments in the format of the `JAMR` aligner since most of the AMR parsers adopt this alignment type.

For supporting our method, we used the Glove[2] embeddings of 100 dimensions pre-trained for the Portuguese language [13] and some lexical resources. We organized our method into three phases carried out over input annotated sentences: preprocessing, mapping, and aligning.

In the first step, we tokenized the sentences and lemmatized each token, applying the Stanza tool [24] trained for Portuguese. The Portuguese tokenization is slightly different from English. For example, some hyphenated words, as "*via-me*" and "*ouvi-la*" (translated for "saw me" and "hear her"), should be separated by the hyphen, whereas other words, as "*segunda-feira*" and "*recém-casados*"

[2] We also experimented other pre-trained models, as Word2Vec [21], Wang2Vec [17], and FastText [5] trained for Portuguese with dimensions of 50, 100, and 300.

(translated for "Monday" and "newly married"), should not be separated. To detail the next steps, we will use Figure 10 as an example.

```
# ::snt Mas Pedro não respondeu
# ::alignments 0-1|0 3-4|0.0 2-3|0.0.0
1-2|0.0.1+0.0.1.0+0.0.1.0.0 5-6|0.0.2 6-7|0.0.2.0

(c / contrast-01                    0
   :ARG2 (r / responder-01          0.0
      :polarity -                   0.0.0
      :ARG0 (p / person             0.0.1
         :name (n / name            0.0.1.0
            :op1 "Pedro"))))        0.0.1.0.0
```

Fig. 10. An example of AMR for the sentence "*Mas Pedro não respondeu*"(But Peter did not answer).

In the next step, we mapped each concept to its respective position in the graph. One can see that we mapped the `contrast-01` concept to the root of the graph `0`, its child `responder-01` to `0.0`, and their children `-` and `person` to their respective positions `0.0.0` and `0.0.1`. To do this, we used the Penman tool [11].

In the last step, we aligned the word tokens of the sentence with the concepts of the graph. The AMR language has two concept types: concrete and abstract (or special keywords) ones. The former are those that are explicitly present in the sentence, while the latter are not. In Figure 10, we can see that `responder-01` is a concrete concept, since it is in the sentence, while the `contrast-01`, `person`, and `name` concepts are abstract[3].

To align concrete concepts, we used the Word Mover's Distance (WMD) [15] and the pre-trained Glove embeddings of 100 dimensions. The WMD is a distance function where the lower distance value indicates a higher similarity to the documents. It measures the minimum amount of distance that embedded words of one document need to "travel" to reach the embedded words of another document.

We used this distance function to evaluate a distance between the embedded word tokens in the sentence and the embedded concepts in the graph to produce alignments with more semantics information than only string-match. Furthermore, we empirically defined a maximum distance (threshold) of 1.5 to match a token with a concept, i.e., our strategy maps a word with a concept only if the distance between them is less than the defined threshold. Figure 11 shows this strategy to align the words of the sentence with concrete concepts of the graph.

From this figure, $W = \{w_1, ..., w_n\}$ is the set of words of a sentence and $C = \{c_1, ..., c_n\}$ is the set of concrete concepts of the graph. Our method aligns a w_i with a c_j if and only if the WMD value between w_i and c_j is lower than 1.5, and that value is the lowest among the other delta values.

[3] "`Pedro`" and `-` are constants, as they get no variable.

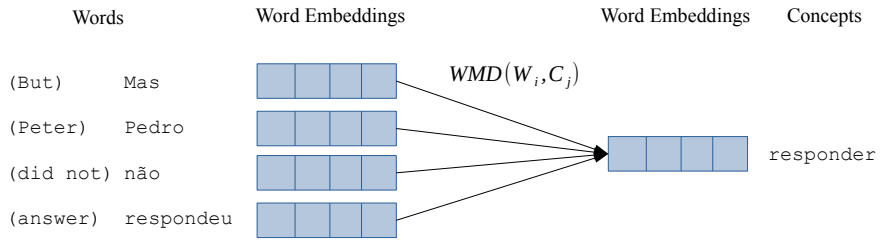


Fig. 11. Aligning word tokens with concrete concepts

To align abstract concepts, we used some lexical resources (list of words) to aid and get a higher recall in the alignment. At this time, the AMR formalism has 44 abstract concepts[4] and 110 concepts that represent named entities[5]. For instance, in Figure 10, `person` is a concept that produces a named entity and, as this concept is in our resource, our alignment strategy aligns this concept and its children with the *Pedro* (Peter) token, which is the span `1-2`. Hence, our method generating the alignment `1-2|0.0.1+0.0.1.0+0.0.1.0.0`, which means that the span `1-2` is aligned with the concept `person` (`0.0.1`) plus `name` (`0.0.1.0`) plus "`Pedro`" (`0.0.1.0.0`).

In addition to named entities and abstract concepts, AMR concepts encompass contrastive conjunctions and negations. To align these concepts, we also created more two lexical resources (list of words) for these concepts, one with constrative words[6] and another with negation words[7]. Likewise, as abstract concepts and named entities, as the words *Mas* (But) and *não* (not) are in our resources, our alignment method aligns the spans `0-1` and `2-3`, which are the words *Mas* (But) and *não* (not), respectively, with nodes `0` and `0.0.0`, which are `contrast-01` and `-`, respectively, producing the alignments `0-1|0` and `2-3|0.0.0` (see Figure 10).

Our alignment tool is available at `http://github.com/rafaelanchieta/AMR-Aligner`. In what follows, we detail our experiments with the aligner and the obtained results.

5 Experiments and Results

We performed two experiments, one intrinsic and another extrinsic. In the first, we randomly chose and manually aligned one hundred sentences with their respective AMRs from the Little Prince corpus [1]. Then, we compared the manual alignment with the alignments produced by Flanigan et al. [10] (henceforth, we refer to it as `JAMR`), Pourdamghani et al. [23] (henceforth, we refer to it as `UNSU`),

[4] `https://amr.isi.edu/doc/amr-dict.html`
[5] `https://www.isi.edu/~ulf/amr/lib/ne-types.html`
[6] `https://www.isi.edu/~ulf/amr/lib/popup/contrast.html`
[7] `https://www.enchantedlearning.com/wordlist/negativewords.shtml`

Liu et al. [18] (henceforth, we refer to it as TAMR), and our proposed aligner (henceforth, we refer to it as Aligner-BR). We converted the alignment format of UNSU to produce alignments in the format of the JAMR aligner. In Table 1, we show the obtained results in the intrinsic evaluation.

Table 1. Results in the intrinsic evaluation

Aligner	Precision	Recall	F-score
JAMR	0.71	0.86	0.78
UNSU	0.48	0.58	0.53
TAMR	0.70	0.88	0.78
Aligner-BR	**0.92**	**0.95**	**0.93**

As we can see, our aligner outperformed those developed for English, which means that our alignment strategy produced alignments more consistent with those manually produced. To get these values, we followed the evaluation method of Flanigan et al. [10].

In order to confirm the intrinsic evaluation results, we performed an extrinsic evaluation. Thus, we adapted the AMR parsers of Damonte et al. [8] (henceforth, we refer to it as AMREager) and Wang et al. [27] (henceforth, we refer to it as CAMR) for the Portuguese language. We chose these parsers because they make use of the alignments, are open source, need only minor modifications for reuse with other languages, and have a good performance on small corpora. They require tokenization, lemmatization, Part-Of-Speech (POS) tagging, chunking parsing, dependency parsing, and Named Entity Recognition (NER), which for Portuguese are provided by Stanza tool [24] (tokenization, lemmatization, POS tagging, and dependency parsing), LX-Parser tool [25] (chunking parsing), and spaCy tool[8] (NER).

We trained these parsers on The Little Prince corpus of the Portuguese language [1], which contains 1,274, 145, and 143 sentences for training, development, and testing, respectively. To compare the results of the parsers, we used the traditional Smatch metric [7] and the more recently proposed SEMA metric [2]. Table 2 shows the obtained results in the extrinsic evaluation.

From this table, one realizes that our aligner improved the adapted AMR parsers for Portuguese in both metrics, confirming the intrinsic evaluation results. Moreover, the CAMR parser achieved a competitive result (50% f-score on the Smatch metric) compared to the RBAMR parser [3] (53% f-score over the same corpus and metric), a rule-based AMR parser designed for the Portuguese language.

We also performed a fine-grained error analysis to identify the weaknesses of our aligner. For that, we used the evaluation tool of Damonte et at. [8] to compare the CAMR parser, as it achieved the best results with the best aligners JAMR, TAMR, and Aligner-BR. We present the obtained results in Table 3.

[8] https://spacy.io

Table 2. Results in the extrinsic evaluation

Parser	Aligner	Smatch			SEMA		
		P	R	F1	P	R	F1
CAMR	JAMR	0.46	0.34	0.39	0.20	0.14	0.16
	UNSU	0.35	0.29	0.32	0.15	0.10	0.12
	TAMR	0.47	0.36	0.41	0.24	0.18	0.20
	Aligner-BR	**0.54**	**0.47**	**0.50**	**0.32**	**0.27**	**0.29**
AMREager	JAMR	0.44	0.33	0.38	0.18	0.13	0.15
	UNSU	0.34	0.27	0.30	0.14	0.09	0.11
	TAMR	0.38	0.34	0.36	0.17	0.15	0.16
	Aligner-BR	**0.51**	**0.45**	**0.48**	**0.30**	**0.26**	**0.28**

Table 3. Fine-grained results

CAMR	JAMR	TAMR	Aligner-BR
Metric	**F-score**		
Unlabeled	0.46	0.48	**0.58**
No WSD	0.41	0.42	**0.52**
NER	0.00	**0.13**	0.00
Wiki	0.00	0.00	0.00
Negations	0.00	0.00	**0.61**
Concepts	0.52	0.53	**0.58**
Reetrancies	0.05	0.06	**0.07**
SRL	0.32	0.37	**0.52**

We can see that the CAMR parser + Aligner-BR outperformed the other aligners in most metrics. The models tied for the Wiki metric due to the corpus not having wiki annotation. In the NER metric, the TAMR aligner performed better than the other aligners. This result is because of the specific rules to align named entities and the Morphosemantic database that this aligner makes use of. Besides, our aligner achieved only 0.07 of reentrancies, as the aligner is not prepared to align reentrancies. One solution could be to model reentrancies as a tree, according to Zhang et al. [28]. Treating this issue remain for future work. We also intend to investigate the Morphosemantic database, aiming to improve the accuracy in the alignment of named entities.

The adapted parsers with our alignment tool and trained on The Little Prince corpus of the Portuguese language are available at GitHub website[9].

[9] CAMR - http://github.com/rafaelanchieta/CAMR-PT
AMREager - http://github.com/rafaelanchieta/amr-eager-pt

6 Conclusion

In this paper, we presented an AMR alignment method designed for the Portuguese language. It is based on pretrained word embeddings and on the Word Mover's Distance for matching word tokens in the sentences and nodes in the corresponding AMR graphs. This simple approach may be adopted for other languages that have few resources, aiming to get tools for natural language understanding tasks. Furthermore, this aligner may be useful to build or increase semantic resources, using a promising approach as back-translation [26]. Future work includes adopting multilingual word embeddings [16] to produce alignments for other languages. More details about AMR resources and tools for the Portuguese language may be found at OPINANDO project[10].

Acknowledge

The authors are grateful to IFPI and USP for supporting this work.

References

1. Anchiêta, R., Pardo, T.: Towards AMR-BR: A SemBank for Brazilian Portuguese language. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation. pp. 974–979. European Languages Resources Association, Miyazaki, Japan (May 2018)
2. Anchiêta, R.T., Cabezudo, M.A.S., Pardo, T.A.S.: SEMA: an extended semantic evaluation metric for amr. In: (To appear) Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing. La Rochelle, France (April 2019)
3. Anchiêta, R.T., Pardo, T.A.S.: A rule-based amr parser for portuguese. In: Simari, G.R., Fermé, E., Gutiérrez Segura, F., Rodríguez Melquiades, J.A. (eds.) Advances in Artificial Intelligence. pp. 341–353. Springer International Publishing, Trujillo, Peru (Nov 2018)
4. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. pp. 178–186. Association for Computational Linguistics, Sofia, Bulgaria (Aug 2013)
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
6. Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19**(2), 263–311 (1993)
7. Cai, S., Knight, K.: Smatch: an evaluation metric for semantic feature structures. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 748–752. Association for Computational Linguistics, Sofia, Bulgaria (Aug 2013)

[10] `https://sites.google.com/icmc.usp.br/opinando/`

8. Damonte, M., Cohen, S.B., Satta, G.: An incremental parser for abstract meaning representation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 536–546. Association for Computational Linguistics, Valencia, Spain (Apr 2017)
9. Fellbaum, C., Osherson, A., Clark, P.E.: Putting semantics into wordnet's "morphosemantic" links. In: Vetulani, Z., Uszkoreit, H. (eds.) Human Language Technology. Challenges of the Information Society. pp. 350–358. Springer Berlin Heidelberg, Berlin, Heidelberg (Oct 2009)
10. Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., Smith, N.A.: A discriminative graph-based parser for the abstract meaning representation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1426–1436. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014)
11. Goodman, M.W.: Penman: An open-source library and tool for AMR graphs. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 312–319. Association for Computational Linguistics, Online (Jul 2020)
12. Gorn, S.: Explicit definitions and linguistic dominoes. In: Systems and Computer Science, Proceedings of the Conference held at Univ. of Western Ontario. pp. 77–115 (1965)
13. Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Silva, J., Aluísio, S.: Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology. pp. 122–131. Sociedade Brasileira de Computação, Uberlândia, Brazil (Oct 2017)
14. Kingsbury, P., Palmer, M.: From TreeBank to PropBank. In: Proceedings of the Third International Conference on Language Resources and Evaluation. pp. 1989–1993. European Language Resources Association, Las Palmas, Canary Islands - Spain (May 2002)
15. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. pp. 957–966. PMLR, Lille, France (July 2015)
16. Lample, G., Conneau, A., Denoyer, L., Ranzato, M.: Unsupervised machine translation using monolingual corpora only. In: International Conference on Learning Representations. Vancouver, Canada (Apr–May 2018)
17. Ling, W., Dyer, C., Black, A.W., Trancoso, I.: Two/too simple adaptations of Word2Vec for syntax problems. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1299–1304. Association for Computational Linguistics, Denver, Colorado (May–Jun 2015)
18. Liu, Y., Che, W., Zheng, B., Qin, B., Liu, T.: An AMR aligner tuned by transition-based parser. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2422–2430. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018)
19. Lyu, C., Titov, I.: AMR parsing as graph prediction with latent alignment. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 397–407. Association for Computational Linguistics, Melbourne, Australia (Jul 2018)
20. Matthiessen, C., Bateman, J.A.: Text generation and systemic-functional linguistics: experiences from English and Japanese. Pinter Publishers (1991)

21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of International Conference on Learning Representations Workshop. Scottsdale, Arizona (May 2013)
22. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (Oct 2014)
23. Pourdamghani, N., Gao, Y., Hermjakob, U., Knight, K.: Aligning English strings with abstract meaning representation graphs. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 425–429. Association for Computational Linguistics, Doha, Qatar (Oct 2014)
24. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 101–108. Association for Computational Linguistics, Online (Jul 2020)
25. Silva, J., Branco, A., Castro, S., Reis, R.: Out-of-the-box robust parsing of portuguese. In: Pardo, T.A.S., Branco, A., Klautau, A., Vieira, R., de Lima, V.L.S. (eds.) Computational Processing of the Portuguese Language. pp. 75–85. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
26. Sobrevilla Cabezudo, M.A., Mille, S., Pardo, T.: Back-translation as strategy to tackle the lack of corpus in natural language generation from semantic representations. In: Proceedings of the 2nd Workshop on Multilingual Surface Realisation. pp. 94–103. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
27. Wang, C., Xue, N., Pradhan, S.: A transition-based algorithm for AMR parsing. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 366–375. Association for Computational Linguistics, Denver, Colorado (May–Jun 2015)
28. Zhang, S., Ma, X., Duh, K., Van Durme, B.: AMR parsing as sequence-to-graph transduction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 80–94. Association for Computational Linguistics, Florence, Italy (Jul 2019)
29. Zhang, S., Ma, X., Duh, K., Van Durme, B.: Broad-coverage semantic parsing as transduction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 3777–3789. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
30. Zhou, J., Xu, F., Uszkoreit, H., Qu, W., Li, R., Gu, Y.: AMR parsing with an incremental joint model. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 680–689. Association for Computational Linguistics, Austin, Texas (Nov 2016)

CHAPTER

9

EXPLORING THE POTENTIALITY OF SEMANTIC FEATURES FOR PARAPHRASE DETECTION

Rafael Torres Anchiêta and Thiago Alexandre Salgueiro Pardo. 2020. “Exploring the Potentiality of Semantic Features for Paraphrase Detection”. In Proceedings of the 14th International Conference on the Computational Processing of Portuguese (PROPOR).

Contribution statement

R.T. Anchiêta participated in conceiving the experiments, developed the research, and contributed in writing the manuscript. T.A.S. Pardo conceived the experiments, helped writing the manuscript, and supervised the project.

Exploring the Potentiality of Semantic Features for Paraphrase Detection

Rafael Torres Anchiêta( ) and Thiago Alexandre Salgueiro Pardo

Interinstitutional Center for Computational Linguistics (NILC),
Institute of Mathematical and Computer Sciences (ICMC),
University of São Paulo (USP), São Carlos, SP, Brazil
rta@usp.br, taspardo@icmc.usp.br

Abstract. Paraphrase is defined as the repetition of something written or spoken using different words. In this paper, we adopt a feature engineering strategy to perform paraphrase detection at the sentence level. In particular, we explore the potentiality of semantic features, as the similarity between two semantic graphs, a distance function between sentences and the cosine similarity between embedded sentences, using them within several machine learning-based classifiers. We evaluate our approach on the ASSIN benchmark corpus and achieve 80.5% of F-score, outperforming some other detection methods for Portuguese.

Keywords: Paraphrase detection · Semantics · Machine learning

1 Introduction

According to Bhagat and Hovy [5], paraphrases are sentences or phrases that convey the same meaning using different wording, i.e., they represent alternative surface forms in the same language expressing the same semantic content of the original forms [19]. For example, sentences 1 and 2 are paraphrases of each other.

1. It is a strange term, but we have got used to it.
2. This term is strange, however, we are accustomed to it.

Automatically detecting paraphrases may be useful for several Natural Language Processing (NLP) tasks, for instance, summarization [15], question answering [20], plagiarism detection [22], semantic parsing [31] and machine translation [29], among others.

Despite the importance of detecting paraphrases, few studies have focused on this task for the Portuguese language. Moreover, the reported achieved results are still far from the ones for the English language. For Portuguese, the achieved results are under 40% of F-score, whereas, for English, the results are over 85%. One possible reason for the few studies in Portuguese is that authors have focused on the related task of textual entailment recognition, which is the task of deciding whether the meaning of one text may be inferred from another one [12].

P. Quaresma et al. (Eds.): PROPOR 2020, LNAI 12037, pp. 228–238, 2020.
https://doi.org/10.1007/978-3-030-41505-1_22

In this paper, we perform feature engineering to develop a new method to identify whether two sentences are paraphrases of each other. In particular, we explore 4 semantic features: Word Mover Distance (WMD) [16], Smooth Inverse Frequency (SIF) [3], the cosine between two embedded sentences (COS), and the similarity between sentences encoded as Abstract Meaning Representation (AMR) graphs [4]. We train some classifiers, as Support Vector Machine (SVM), Naïve Bayes (NB), Decision Trees (DT), Neural Networks (NN) and Logistic Regression (LR), and evaluate them on the ASSIN benchmark corpus [11], reaching the best F-score of 80.5% with the SVM classifier, outperforming some other methods for Portuguese.

The remaining of this paper is organized as follows. Section 2 describes the main related work. In Sect. 3, we present the used corpus. Section 4 details our paraphrase identification method and the adopted features. In Sect. 5, we report our experiments and the achieved results. Finally, Sect. 6 concludes the paper, indicating future research.

2 Related Work

For the Portuguese language, there are few approaches that strictly tackle the paraphrase detection task. At this point, it is important to let clear that there is a subtle difference between paraphrase detection and similarity/entailment identification, as claimed in [30]. For the latter task, other works do exist. Following [30], we specifically focus on the former approaches.

Cordeiro et al. [10] developed a metric named *Sumo-Metric* for semantic relatedness between two sentences based on the overlapping of lexical units. Although the authors evaluated their metric on a corpus for the English language, the metric is language-independent.

Rocha and Cardoso [28] modeled the task as a supervised machine learning problem. However, they handled the issue as a multi-class task, classifying sentence pairs into entailment, none, or paraphrase. Thus, they employed a set of features of the lexical, syntactic, and semantic levels to represent the sentences in numerical values, and fed these features into some machine learning algorithms. They evaluated their method on the training set of the ASSIN corpus, using both European and Portuguese partitions. The method obtained 0.52 of F-score using a SVM classifier.

Souza and Sanches [30] also dealt with the problem with supervised machine learning. However, their objective was to explore sentence embeddings for this task. They used a pre-trained FastText model [7] and the following features: the average of the vectors, the value of Smooth Inverse Frequency (SIF), and weighted aggregation based on Inverse Document Frequency (IDF). With these features, their method reached 0.33 of F-score using a SVM classifier on balanced data of the ASSIN corpus for European and Portuguese partitions.

Consoli et al. [9] analyzed the capabilities of the coreference resolution tool CORP [13] for identification of paraphrases. The authors used CORP to identify noun phrases that may help to detect paraphrases between sentence pairs. They

evaluated their method on 116 sentence pairs from the ASSIN corpus, achieving 0.53 of F-score.

For the English language, recent works have focused on deep learning models [17,18,32] because of the availability of large corpora. These works achieve F-scores over 85%.

3 The Corpus

To evaluate our model, we used the ASSIN corpus [11]. It contains 10,000 sentence pairs, 5,000 written in Brazilian Portuguese and 5,000 in European Portuguese. Each language has 2,500, 500, and 2,000 pairs for training, development, and testing, respectively, as shown in Table 1.

Table 1. Organization of the ASSIN corpus

Language	Training	Development	Testing
Brazilian Portuguese	2,500	500	2,000
European Portuguese	2,500	500	2,000

The data in the corpus is organized into three categories: entailment, none, and paraphrase. Table 2 presents the distribution of these categories in the corpus. As we can see, the corpus is unbalanced concerning the paraphrase label, since the proportion of entailment and none labels is much higher than paraphrases. For example, the none label has 73.16% of examples in the corpus and the entailment label has 20.80%. Together, they sum 93.96% of the examples in the corpus.

Table 2. Distribution of labels in the ASSIN corpus

Label	Brazilian Port.			European Port.			Total #	Proportion %
	Train.	Dev.	Test.	Train.	Dev.	Test.		
Entailment	437	92	341	613	116	481	2,080	20.80
None	1,947	384	1,553	1,708	338	1,386	7,316	73.16
Paraphrase	116	24	106	179	46	133	604	6.04

In this paper, we used both European and Brazilian Portuguese languages for paraphrase detection. In Table 3, we show examples of paraphrase pairs in the corpus.

Table 3. Examples of paraphrase pairs for Portuguese

Language Variety	Sentences
Brazilian	(1) *De acordo com o site TMZ, a cantora Britney Spears comprou uma nova casa*
	(2) *Segundo informações divulgadas pelo site TMZ, Britney Spears está de casa nova* (In English, "According to the TMZ website, singer Britney Spears bought a new home.")
European	(3) *Hitler não queria exterminar os judeus na época, ele queria expulsar os judeus*
	(4) *Naquela altura, Hitler não queria exterminar os judeus mas sim expulsá-los* (In English, "Hitler did not want to exterminate the Jews, but to expel them.")

4 Paraphrase Identification Method

Although the ASSIN corpus has three labels, we joined the entailment and none labels into one unique label named "non-paraphrase", which is our negative class. Table 4 shows the new configuration of the ASSIN corpus.

Table 4. New distribution of labels in the ASSIN corpus

Label	Brazilian Port.			European Port.			Total #	Proportion %
	Train.	Dev.	Test.	Train.	Dev.	Test.		
Non-paraphrase	2,384	476	1,894	2,321	454	1,867	9,396	93.96
Paraphrase	116	24	106	179	46	133	604	6.04

We did this modification to formulate the task as a binary classification problem, since we aim to identify whether a sentence pair shows a paraphrase or not. Thus, we formulate the problem in the following way. Let S be a set of sentence pairs. We provide input data in the form of $(x_1^{(i)}, x_2^{(i)}, b^{(i)})$ for $i \in [n]$, where n is the number of training sentences, $x_1^{(i)}$ and $x_2^{(i)}$ are the input sentences, and $b^{(i)} \in \{0, 1\}$ indicates a binary classification that informs whether $x_1^{(i)}$ and $x_2^{(i)}$ are paraphrases of each other. In summary, the aim is to learn a classifier c that, given unseen sentence pairs, classifies whether they are paraphrases, as in Eq. 1.

$$c : S \times S \rightarrow 0, 1 \tag{1}$$

To classify these sentence pairs, we extract some semantic features. We focused on semantic features because detecting paraphrases involves understanding the meaning of the sentences. Some of the adopted features need to be modeled as word embeddings, which are vectors of real-valued numbers that represent

particular words. As the ASSIN corpus is small to train embedding models, we used pre-trained word embeddings [14] and evaluated different models such as Word2Vec [23], FastText [7], and Glove [25] with dimensions of 50, 100, and 300. For Wor2Vec and FastText, we analyzed two training methods: Skip-Gram and CBOW. We describe the extracted features in what follows.

Word Mover Distance (WMD). It is a feature that assesses the distance between two documents even when they have no words in common [16]. It measures the dissimilarity between two text documents as the minimum amount of distance that embedded words of one document need to "travel" to reach the embedded words of another document. It is important to notice that WMD is a distance function, i.e., the lower the distance value is, the more similar the documents are. For getting the WMD distance, we first tokenized and removed stopwords of the sentences, using the Natural Language Toolkit (NLTK) [6]; next, we got the embeddings for the words of the sentences; finally, to get the WMD distance, we used the method from Gensim library [27] that receives a sentence pair encoded as word embeddings as input and returns the WMD value.

Cosine of Word Embeddings (COS). For calculating the cosine similarity between sentences, we got the embeddings for the words, computed the average of the word embeddings for each sentence, and calculated the cosine similarity between these vectors, applying Eq. 2, where $\overrightarrow{u} \cdot \overrightarrow{x}$ is the dot product of the two vectors.

$$\cos\theta = \frac{\overrightarrow{u} \cdot \overrightarrow{x}}{\|\overrightarrow{u}\|\|\overrightarrow{x}\|} \tag{2}$$

Smooth Inverse Frequency (SIF). Computing the average of the word embeddings, as in the cosine similarity, tends to give too much weight to words that may be irrelevant [3]. SIF tries to solve this problem, giving more weight to words that contribute to the semantics of the sentence. For this purpose, Eq. 3 is used, where a is a hyper-parameter set to 0.001 by default and $p(w)$ is the estimated word frequency in the corpus.

$$SIF(w) = \frac{a}{(a + p(w))} \tag{3}$$

Abstract Meaning Representation (AMR). It is a semantic representation language designed to capture the meaning of a sentence [4]. This language represents sentences as directed acyclic graphs, where the nodes are concepts and edges represent the relation among them, explicitly showing semantic features, as semantic roles, word sense disambiguation, negation, and others. Therefore, sentences with the same meaning should result in similar graphs. For instance, the sentences "James did not see the girl who wanted him." and "James did not see the girl who he was wanted by." may be encoded as the AMR graph

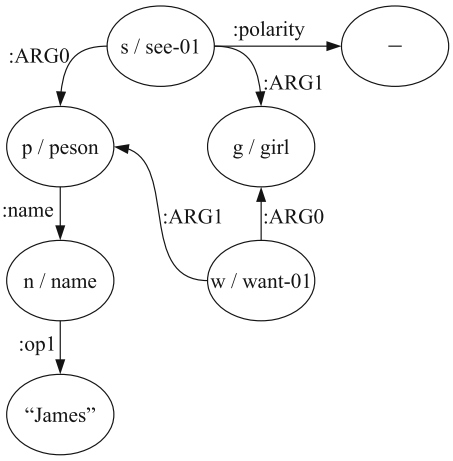


Fig. 1. An example of AMR graph. The `see-01` node is the root of the graph, the `person` node indicates a named entity, and the `polarity` relation illustrates a negation. One may also see PropBank-like semantic roles. For more details, we suggest consulting the original AMR paper [4].

shown in Fig. 1. To take this information into account, we parsed the sentence pairs into AMR graphs using a rule-based AMR parser [1] and computed the similarity between the graphs using the SEMA metric [2].

Following the feature extraction step, we used some classifiers to evaluate our approach. We used Support Vector Machine (SVM), Naïve Bayes (NB), Decision Trees (DT), Neural Networks (NN), and Logistic Regression (LR) in the Scikit-Learn library [24]. In what follows, we detail our experiments and the obtained results.

5 Experiments and Results

After extracting the features, we used the ASSIN corpus with two labels, as shown in Table 4, to evaluate our approach. As we can see, the corpus is very unbalanced concerning the paraphrase label. This causes difficulties to the learning, since a classifier will learn much more about non-paraphrase information. To mitigate this issue, we applied the SMOTE (Synthetic Minority Over-sampling Technique) technique [8] to balance the data. It creates synthetic data for the minority class in order to obtain a balanced corpus.

After evaluating different pre-trained word embeddings with the extracted features to feed the classifiers, the best setting was reached with the SVM classifier with linear kernel and the Word2Vec model of 300 dimensions using the Skip-Gram as training method. Table 5 shows the obtained results with that setting in both unbalanced and balanced version of the corpus.

To compare with our approach, we developed a baseline method based on the word overlap between the sentence pairs. First, we tokenized the sentence pairs; next, we computed the number of tokens in the intersection between the sentence pairs; we, then, applied the SMOTE technique to balance the data; and,

Table 5. Results of the SVM classifier for paraphrase detection

	Class	Precision	Recall	F-score
Unbalanced	Paraphrase	0.20	0.81	0.32
	Non-paraphrase	0.98	0.80	0.88
Balanced	Paraphrase	0.80	0.82	0.81
	Non-paraphrase	0.81	0.80	0.80

at last, we trained the SVM classifier. In addition to the baseline method, we also compared our approach with the method of Souza and Sanches [30], since they used the same corpus and oversampling technique. In Table 6, we present a comparison among the approaches.

Table 6. Comparison among approaches for paraphrase identification

Method	F-score
Souza and Sanches [30]	0.333
Baseline	0.730
Our approach	**0.805**

One may see that our approach outperforms the related work, achieving results close to those for the English language. Moreover, using the same pre-trained word embeddings and classifier above, we investigated the importance of each feature in the classification, as depicted in Fig. 2.

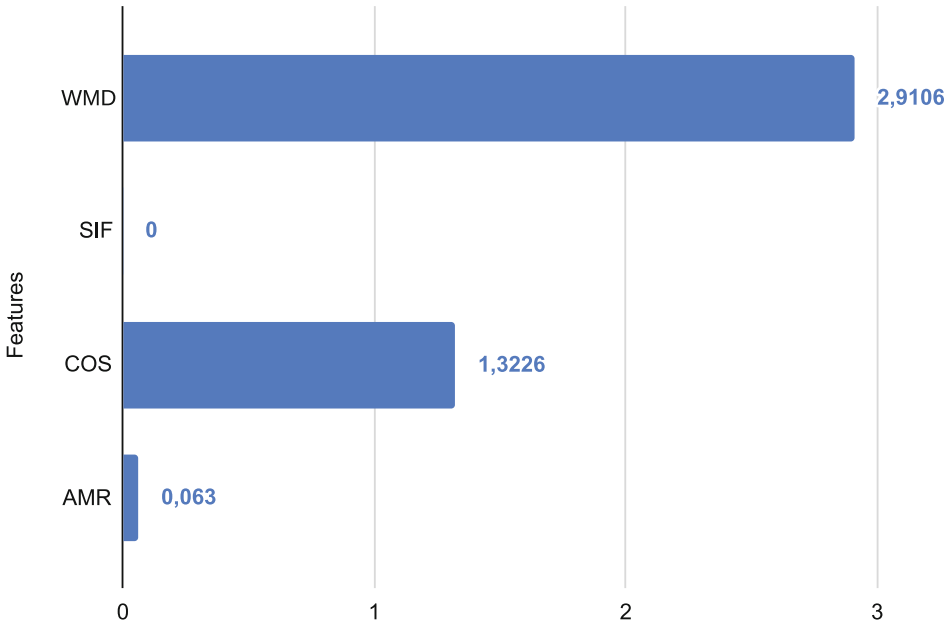


Fig. 2. Importance of each feature

Overall, the features WMD, COS, and AMR are more relevant for classification, while SIF did not contribute to it. We believe that the SIF feature fails because the corpus is small and this metric uses information about the estimated frequency of words.

We performed an ablation study aiming to investigate how each feature may improve the classification. Table 7 presents the results for the study using the SVM classifier. We can see that the WMD + COS and WMD + COS + AMR features achieved the best F-score values (0.805). However, the AMR feature had little contribution to classification, since the WMD + COS reached higher value.

Table 7. Ablation study with the features for the SVM classifier

Features	F-score
WMD	0.800
COS	0.740
AMR	0.580
WMD + COS	**0.805**
WMD + AMR	0.803
WMD + COS + AMR	**0.805**

Another conclusion is that the COS feature alone produced better results than the AMR feature alone. This is relevant because such measures follow different semantic paradigms: COS is based on the word embeddings, which "implicitly" represent semantics, while AMR "explicitly" indicates the semantic constituents of the text passage of interest. The achieved result may reflect that embedding learning is currently more robust than AMR learning/parsing, but may also indicate that our feature computation did not take full advantage of AMR potentiality. This issue remains for future exploration.

We performed a detailed error analysis and could realize that automatically distinguishing paraphrase from entailment cases is not a trivial task, as a paraphrase may be viewed as a mutual (or bidirectional) entailment. Textual entailment is the relationship between a text T and a hypothesis H, where $T \rightarrow H$ (T entails H), while a paraphrase may be viewed as $T \rightarrow H$ and $H \rightarrow T$. Because of this, SVM misclassified the sentence "*Segundo Lagarde, esse fenômeno deverá levar o FMI a revisar para baixo a previsão de crescimento.*" and "*Esse fenômeno deve levar o FMI a revisar para baixo as projeções de crescimento.*" as paraphrases, whereas it also missclassified the sentences "*Nunca antes um pontífice havia ido ao plenário das Casa Legislativas dos Estados Unidos.*" and "*Ele será o primeiro papa no plenário da Casa Legislativa dos Estados Unidos.*" as non-paraphrases.

Treating such subtleties in classification and investigating which types of paraphrases were not identified remain for future work.

6 Final Remarks

In this paper, we explored the potentiality of semantic features in a machine learning solution to detect paraphrases for the Portuguese Language. We evaluated our approach on the ASSIN benchmark corpus, achieving 80.5% of F-score, outperforming some reported results on the literature for Portuguese.

Future work includes exploring other semantic features and classification strategies, looking for more data to run deep learning methods. A new feature that may be promising to the task comes from discourse structuring studies, in particular, from the works on parsing texts according to the Cross-document Structure Theory [26], which is a discourse model that predicts some paraphrase-like relations among text passages. Discourse parsers for Portuguese already exist (see, e.g., [21]) and might be used.

To the interested reader, the source code and trained models that we tested are available online (http://github.com/rafaelanchieta/paraphrase-detection). Additional information about this work may be found at the OPINANDO project webpage (https://sites.google.com/icmc.usp.br/opinando/).

Acknowledgements. The authors are grateful to *Instituto Federal do Piauí* (IFPI) and USP Research Office (PRP 668) for supporting this work.

References

1. Anchiêta, R.T., Pardo, T.A.S.: A rule-based AMR parser for portuguese. In: Simari, G.R., Fermé, E., Gutiérrez Segura, F., Rodríguez Melquiades, J.A. (eds.) IBERAMIA 2018. LNCS (LNAI), vol. 11238, pp. 341–353. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03928-8_28
2. Anchiêta, R.T., Cabezudo, M.A.S., Pardo, T.A.S.: SEMA: an extended semantic evaluation metric for amr. In: (To appear) Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (2019)
3. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: Proceeding of the 5th International Conference on Learning Representations (2017)
4. Banarescu, L., et al.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pp. 178–186 (2013)
5. Bhagat, R., Hovy, E.: What is a paraphrase? Comput. Linguist. **39**(3), 463–472 (2013)
6. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc., Sebastopol (2009)
7. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
9. Consoli, B.S., Neto, J.F.S., de Abreu, S.C., Vieira, R.: Análise da capacidade de identificação de paráfrase em ferramentas de resolução de correferência. Linguamática **10**(2), 45–51 (2018)

10. Cordeiro, J., Dias, G., Brazdil, P.: A metric for paraphrase detection. In: International Multi-Conference on Computing in the Global Information Technology, pp. 1–7. IEEE (2007)
11. Fonseca, E., Santos, L., Criscuolo, M., Aluísio, S.: Assin: Avaliação de similaridade semântica e inferência textual. In: Proceedings of the 12th International Conference on the Computational Processing of Portuguese, pp. 13–15 (2016)
12. Fonseca, E.R., dos Santos, L.B., Criscuolo, M., Aluísio, S.M.: Visão geral da avaliação de similaridade semântica e inferência textual. Linguamática **8**(2), 3–13 (2016)
13. Fonseca, E., Sesti, V., Antonitsch, A., Vanin, A., Vieira, R.: Corp: Uma abordagem baseada em regras e conhecimento semântico para a resoluçao de correferências. Linguamática **9**(1), 3–18 (2017)
14. Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Silva, J., Aluísio, S.: Portuguese word embeddings: evaluating on word analogies and natural language tasks. In: Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology, pp. 122–131 (2017)
15. Jing, H., McKeown, K.R.: Cut and paste based text summarization. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pp. 178–185. Association for Computational Linguistics (2000)
16. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 957–966 (2015)
17. Lan, W., Xu, W.: Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3890–3902 (2018)
18. Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4487–4496. Association for Computational Linguistics (2019)
19. Madnani, N., Dorr, B.J.: Generating phrasal and sentential paraphrases: a survey of data-driven methods. Comput. Linguist. **36**(3), 341–387 (2010)
20. Marsi, E., Krahmer, E.: Explorations in sentence fusion. In: Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05) (2005)
21. Maziero, E.G., del Rosário Castro Jorge, M.L., Pardo, T.A.S.: Revisiting cross-document structure theory for multi-document discourse parsing. Inf. Process. Manag. **50**(2), 297–314 (2014)
22. McClendon, J.L., Mack, N.A., Hodges, L.F.: The use of paraphrase identification in the retrieval of appropriate responses for script based conversational agents. In: Proceedings of the 27th International Flairs Conference, pp. 196–201 (2014)
23. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of International Conference on Learning Representations Workshop (2013)
24. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
25. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)

26. Radev, D.: A common theory of information fusion from multiple text sources step one: cross-document structure. In: Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue, pp. 74–83. Association for Computational Linguistics, Hong Kong, China, October 2000
27. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50 (2010)
28. Rocha, G., Lopes Cardoso, H.: Recognizing textual entailment and paraphrases in Portuguese. In: Oliveira, E., Gama, J., Vale, Z., Lopes Cardoso, H. (eds.) EPIA 2017. LNCS (LNAI), vol. 10423, pp. 868–879. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65340-2_70
29. Sekizawa, Y., Kajiwara, T., Komachi, M.: Improving Japanese-to-English neural machine translation by paraphrasing the target language. In: Proceedings of the 4th Workshop on Asian Translation (WAT2017), pp. 64–69 (2017)
30. Souza, M., Sanches, L.M.P.: Detecção de paráfrases na língua portuguesa usando sentence embeddings. Linguamática **10**(2), 31–44 (2018)
31. Su, Y., Yan, X.: Cross-domain semantic parsing via paraphrasing. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1235–1246 (2017)
32. Tomar, G.S., Duque, T., Täckström, O., Uszkoreit, J., Das, D.: Neural paraphrase identification of questions with noisy pretraining. In: Proceedings of the First Workshop on Subword and Character Level Models in NLP, pp. 142–147 (2017)

CHAPTER

10

CONCLUSION

10.1 Concluding remarks

Abstract Meaning Representation (AMR) is a semantic representation language that may be encoded as a rooted direct acyclic graph. To produce an AMR structure, an annotated corpus, an evaluation tool, and, in most cases, an alignment method are necessary. This work covers these three requirements to generate and/or adapt AMR parsers. First, an annotated AMR corpus was created. Next, AMR parsing strategies for Portuguese were developed and adapted. Then, an alignment strategy and an evaluation method to support the produced AMR parsers were designed. Finally, these produced resources, methods, and tools were evaluated on the paraphrase detection task.

In this way, one may conclude that our hypotheses were confirmed, since it was possible to map the surface textual into AMR structures with some accuracy. Also, it was possible to improve AMR parsers, improving the alignment between nodes and tokens, and adopting a more robust evaluation metric in the training phase of AMR parsers. In what follows, each topic of the research is commented on.

10.1.1 Annotation process

In Chapter 3, the creation process of the first annotated AMR corpus for Brazilian Portuguese is presented. In addition to the annotated corpus itself, an align-based approach to map an annotation from English to Portuguese is detailed. This approach takes on the advantage to accelerate the annotation process, saving time and effort for that task. Align-based methods also were adopted to annotate corpora in other languages, as Chinese (LI *et al.*, 2016) and Spanish (MIGUELES-ABRAIRA; AGERRI; ILARRAZA, 2018). Moreover, our paper points out some difficult phenomena to annotate, motivating the construction of a new corpus (CABEZUDO; PARDO, 2019) and annotation strategies (CABEZUDO; MILLE; PARDO, 2019). These *hard*

case phenomena already being addressed by the OPINANDO[1] project researches. The annotated corpus is available at <https://github.com/rafaelanchieta/amr-br>.

10.1.2 AMR parsing methods

In Chapter 4, the paper introduces the first AMR parser for Portuguese. It is based on manually designed rules from preprocessed sentences with syntactic and semantic information. This approach explores the hypothesis that an AMR structure is similar to the a dependency tree. Thus, six rules for converting a preprocessed sentence into an AMR graph were proposed. Besides this rule-based method, the paper extends the evaluation tool, analyzing the length of the sentences aiming to understand the strengths and mainly the weakness of the parsing method. The developed method achieves a Smatch f-score of 53.3% on average, outperforming a cross-lingual parsing method and reaching a close result to the first AMR parser for English. The parser is available at <https://github.com/rafaelanchieta/rbamr>.

In Chapter 7, the rule-based AMR parser is extended adding rules to it, aiming to deal with the verb 'to be' and contrastive conjunctions. More than that, a pruning method for removing redundant nodes in the graph is implemented. These enhancements have improved and produced competitive an AMR parser for Portuguese. In addition to the rule-based parser, other parsers were adapted and improved for Portuguese. The paper presents three adapted different parsing strategies in order to compare them against the rule-based parser. Furthermore, the paper introduces a new manner to handle reentrancy relations in the adapted parsers and the use of the SEMA metric in the training phase, helping a parsing method to choose better AMR graphs, and, hence improving the scores in the evaluation tool. At last, a detailed error analysis to identify the points to be improved is performed. The adapted parsers are available at <https://github.com/rafaelanchieta/>.

10.1.3 AMR evaluation and alignment

In Chapter 5, a new metric, named SEMA, to evaluate AMR structures is detailed. It extends the Smatch metric, taking into account the dependency of the elements in the graph. The SEMA metric was designed based on a fine-grained analysis of the shortcomings of the Smatch metric, turning the SEMA fairer, faster, and more robust than Smatch. Besides, the SEMA metric is, on average, 8% stricter than the Smatch metric. Furthermore, in Chapter 6, a study on three evaluation metrics, aiming to investigate which metric is more related to human judgment and more adequate to evaluate AMR structures, was carried out. From this study, one found out that SEMA is the more consistent evaluation metric, being more adequate and fairer in relation to human judgment. The SEMA metric is available at <https://github.com/rafaelanchieta/sema>.

[1] <https://sites.google.com/icmc.usp.br/opinando/>

In Chapter 8, the paper presents a new alignment method to link the tokens of the sentence with the nodes in the graph. The alignment strategy adopts a pre-trained word embedding model as a semantic resource to produce word-concept pairs more semantically matched instead of string-match as JAMR aligner (FLANIGAN *et al.*, 2014). This alignment method was evaluated intrinsically over 100 annotated sentences of the AMR-BR corpus and extrinsically in the AMR parsing task. The aligner improved the adapted AMR parsers, achieving a close result to the rule-based AMR parser, which was designed for Portuguese. This aligner may be useful for creating or increasing new AMR corpora using the back-translation approach (CABEZUDO; MILLE; PARDO, 2019), which is a promising method to translate a sentence of interest. The alignment is available at <https://github.com/rafaelanchieta/AMR-Aligner>.

10.1.4 Paraphrase detection

In Chapter 9, the paper details usage of the developed resources and tools for the AMR formalism in the paraphrase detection task. This paper presents four semantic features (word move distance, smooth inverse frequency, cosine similarity, and abstract meaning representation) that feed a binary classifier responsible to identify if two sentences are paraphrases each other. Moreover, an analysis of the implicit and explicit semantic features in the classification is introduced. Although the implicit features contributed more for classification than the explicit features, a fine-grained investigation on the explicit information is necessary. The paraphrase detection method is available at <https://github.com/rafaelanchieta/paraphrase-detection>.

10.2 Limitations

The major limitation of this work is the size of the AMR-BR corpus. It has 1,527 sentences aligned with the 1,562 sentences from its English counterpart, being the former organized into 1,274, 145, and 143 sentences for training, development, and testing, respectively. This limitation directly affects the performance of the AMR parsing task, since it restricts the use of more robust machine learning strategies. For English, for example, the best AMR parser reaches 77% of Smatch F-score. Consequently, the low score of the AMR parsers influences applications that depend on them, producing less powerful tools.

Other limitations are related to AMR parsing methods. The rule-based method treats only the most frequent relations of the AMR-BR corpus, and the adapted parsers have difficulty learning reentrancy relations due to alignment problems. More robust parsers use align-free strategies, however, for that, they require a large annotated corpus.

10.3 Future works

In addition to the creation of large annotated corpora (which already in progress through OPINANDO project), some topics could be investigated in future works.

1. To treat reentrancy relations. In Chapter 7, a simple method to deal with these relations was proposed, showing improvements in the performance of the parsers. Zhang *et al.* (2019a) tackled this issue using an attention-based model that also improved the performance of its parser.

2. To explore graph transduction methods for the AMR parsing task. Transductive learning refers to predicting specific examples given examples from a domain (VAPNIK, 2013). This strategy has achieved state of the art results in that task (ZHANG *et al.*, 2019b).

3. To investigate the SEMA evaluation tool and the aligner tool in other AMR parsing methods, as these tools have improved the performance of the AMR parsers detailed in chapters 7 and 8, respectively.

BIBLIOGRAPHY

ABEND, O.; RAPPOPORT, A. Universal conceptual cognitive annotation (UCCA). In: **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 228–238. Citations on pages 18 and 27.

____. The state of the art in semantic representation. In: **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Vancouver, Canada: Association for Computational Linguistics, 2017. p. 77–89. Citations on pages 17 and 25.

ABZIANIDZE, L.; BOS, J. Thirty musts for meaning banking. In: **Proceedings of the First International Workshop on Designing Meaning Representations**. Florence, Italy: Association for Computational Linguistics, 2019. p. 15–27. Citation on page 17.

ANCHIÊTA, R.; PARDO, T. Towards AMR-BR: A SemBank for Brazilian Portuguese language. In: **Proceedings of the Eleventh International Conference on Language Resources and Evaluation**. Miyazaki, Japan: European Languages Resources Association, 2018. p. 974–979. Citation on page 22.

ANCHIÊTA, R. T.; CABEZUDO, M. A. S.; PARDO, T. A. S. SEMA: an extended semantic evaluation metric for amr. In: **(To appear) Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing**. La Rochelle, France: Springer International Publishing, 2019. Citation on page 22.

ANCHIÊTA, R. T.; PARDO, T. A. S. A rule-based amr parser for portuguese. In: **Ibero-American Conference on Artificial Intelligence**. Trujillo, Peru: Springer International Publishing, 2018. p. 341–353. Citations on pages 20 and 22.

____. Exploring the potentiality of semantic features for paraphrase detection. In: **International Conference on Computational Processing of the Portuguese Language**. Evora, Portugal: Springer International Publishing, 2020. p. 228–238. Citation on page 23.

____. Semantically inspired amr alignment for the portuguese language. In: **(To appear) Proceedings of the 9th Brazilian Conference on Intelligent Systems**. Rio Grande, Brazil: Springer International Publishing, 2020. Citation on page 23.

BANARESCU, L.; BONIAL, C.; CAI, S.; GEORGESCU, M.; GRIFFITT, K.; HERMJAKOB, U.; KNIGHT, K.; KOEHN, P.; PALMER, M.; SCHNEIDER, N. Abstract Meaning Representation for sembanking. In: **Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse**. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 178–186. Citations on pages 18 and 25.

BLACKBURN, P.; BOS, J. Computational semantics. **Theoria: An International Journal for Theory, History and Foundations of Science**, JSTOR, p. 27–45, 2003. Citation on page 17.

BOS, J. Squib: Expressive power of abstract meaning representations. **Computational Linguistics**, v. 42, n. 3, p. 527–535, Sep. 2016. Citation on page 18.

CABEZUDO, M. A. S.; MILLE, S.; PARDO, T. Back-translation as strategy to tackle the lack of corpus in natural language generation from semantic representations. In: **Proceedings of the 2nd Workshop on Multilingual Surface Realisation**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 94–103. Citations on pages 133 and 135.

CABEZUDO, M. A. S.; PARDO, T. Towards a general abstract meaning representation corpus for Brazilian Portuguese. In: **Proceedings of the 13th Linguistic Annotation Workshop**. Florence, Italy: Association for Computational Linguistics, 2019. p. 236–244. Citation on page 133.

CAI, S.; KNIGHT, K. Smatch: an evaluation metric for semantic feature structures. In: **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics**. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 748–752. Citations on pages 20 and 28.

CONDORI, R. E. L.; PARDO, T. A. S. Opinion summarization methods: Comparing and extending extractive and abstractive approaches. **Expert Systems with Applications**, Elsevier, v. 78, p. 124–134, 2017. Citation on page 21.

DAMONTE, M.; COHEN, S. B. Cross-lingual abstract meaning representation parsing. In: **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 1146–1155. Citation on page 22.

DAMONTE, M.; COHEN, S. B.; SATTA, G. An incremental parser for abstract meaning representation. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics**. Valencia, Spain: Association for Computational Linguistics, 2017. p. 536–546. Citations on pages 21, 22, and 28.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Citation on page 19.

FLANIGAN, J.; THOMSON, S.; CARBONELL, J.; DYER, C.; SMITH, N. A. A discriminative graph-based parser for the abstract meaning representation. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics**. Baltimore, Maryland: Association for Computational Linguistics, 2014. p. 1426–1436. Citations on pages 21 and 135.

GOODMAN, J.; VLACHOS, A.; NARADOWSKY, J. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In: **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 1–11. Citations on pages 19 and 21.

HARDY, H.; VLACHOS, A. Guided neural language generation for abstractive summarization using abstract meaning representation. In: **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 768–773. Citations on pages 19 and 21.

HOCKENMAIER, J.; STEEDMAN, M. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn treebank. **Computational Linguistics**, v. 33, n. 3, p. 355–396, 2007. Citation on page 27.

JURAFSKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition**. [S.l.]: Prentice Hall, 2009. ISBN 9780133252934. Citation on page 17.

KAMP, H.; REYLE, U. **From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory**. [S.l.]: Springer Science & Business Media, 1993. Citation on page 26.

KINGSBURY, P.; PALMER, M. From TreeBank to PropBank. In: **Proceedings of the Third International Conference on Language Resources and Evaluation**. Las Palmas, Canary Islands - Spain: European Language Resources Association, 2002. Citations on pages 18 and 25.

KIPPER, K.; KORHONEN, A.; RYANT, N.; PALMER, M. Extensive classifications of english verbs. In: **Proceedings of the 12th EURALEX International Congress**. Torino, Italy: Edizioni dell'Orso, 2006. p. 5–2. Citation on page 26.

KUHLMANN, M.; OEPEN, S. Squibs: Towards a catalogue of linguistic graph Banks. **Computational Linguistics**, v. 42, n. 4, p. 819–827, Dec. 2016. Citation on page 27.

LANGKILDE, I.; KNIGHT, K. Generation that exploits corpus-based statistical knowledge. In: **36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1**. Montreal, Quebec, Canada: Association for Computational Linguistics, 1998. p. 704–710. Citation on page 25.

LEHMANN, F. **Semantic networks in artificial intelligence**. [S.l.]: Elsevier Science Inc., 1992. ISBN 0080420125. Citation on page 17.

LI, B.; WEN, Y.; QU, W.; BU, L.; XUE, N. Annotating the little prince with Chinese AMRs. In: **Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 7–15. Citation on page 133.

LIAO, K.; LEBANOFF, L.; LIU, F. Abstract meaning representation for multi-document summarization. In: **Proceedings of the 27th International Conference on Computational Linguistics**. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 1178–1190. Citation on page 19.

LIU, F.; FLANIGAN, J.; THOMSON, S.; SADEH, N.; SMITH, N. A. Toward abstractive summarization using semantic representations. In: **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Denver, Colorado: Association for Computational Linguistics, 2015. p. 1077–1086. Citation on page 19.

LYU, C.; TITOV, I. AMR parsing as graph prediction with latent alignment. In: **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics**. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 397–407. Citation on page 21.

MARCUS, M. P.; SANTORINI, B.; MARCINKIEWICZ, M. A. Building a large annotated corpus of English: The Penn Treebank. **Computational Linguistics**, v. 19, n. 2, p. 313–330, 1993. Citation on page 26.

MATTHIESSEN, C.; BATEMAN, J. A. **Text generation and systemic-functional linguistics: experiences from English and Japanese**. [S.l.]: Pinter Publishers, 1991. Citation on page 18.

MAY, J. SemEval-2016 task 8: Meaning representation parsing. In: **Proceedings of the 10th International Workshop on Semantic Evaluation**. San Diego, California: Association for Computational Linguistics, 2016. p. 1063–1073. Citation on page 21.

MAY, J.; PRIYADARSHI, J. SemEval-2017 task 9: Abstract meaning representation parsing and generation. In: **Proceedings of the 11th International Workshop on Semantic Evaluation**. Vancouver, Canada: Association for Computational Linguistics, 2017. p. 536–545. Citation on page 21.

MIGUELES-ABRAIRA, N.; AGERRI, R.; ILARRAZA, A. Diaz de. Annotating abstract meaning representations for Spanish. In: **Proceedings of the Eleventh International Conference on Language Resources and Evaluation**. Miyazaki, Japan: European Language Resources Association (ELRA), 2018. p. 3074–3078. Citation on page 133.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. In: **Proceedings of International Conference on Learning Representations Workshop**. Scottsdale, Arizona: [s.n.], 2013. Citation on page 19.

MILLER, G. A. **WordNet: An electronic lexical database**. [S.l.]: MIT press, 1998. Citation on page 26.

MITRA, A.; BARAL, C. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In: **Thirtieth AAAI Conference on Artificial Intelligence**. Phoenix, Arizona, USA: AAAI, 2016. p. 2779–2785. Citation on page 19.

NOORD, R. van; BOS, J. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. **Computational Linguistics in the Netherlands Journal**, v. 7, p. 93–108, Dec 2017. Citations on pages 21 and 22.

OEPEN, S.; ABEND, O.; HAJIC, J.; HERSHCOVICH, D.; KUHLMANN, M.; O'GORMAN, T.; XUE, N. (Ed.). **Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning**. Hong Kong: Association for Computational Linguistics, 2019. Citation on page 21.

PALMER, M.; GILDEA, D.; KINGSBURY, P. The proposition bank: An annotated corpus of semantic roles. **Computational Linguistics**, v. 31, n. 1, p. 71–106, 2005. Citations on pages 25 and 26.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Citation on page 19.

RIBEIRO, L. F. R.; GARDENT, C.; GUREVYCH, I. Enhancing AMR-to-text generation with dual graph representations. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 3181–3192. Citation on page 19.

SACHAN, M.; XING, E. Machine comprehension using rich semantic representations. In: **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 486–492. Citation on page 19.

SAEED, J. **Semantics**. [S.l.]: Wiley, 2016. (Introducing Linguistics). ISBN 9781118430163. Citation on page 17.

SONG, L.; GILDEA, D.; ZHANG, Y.; WANG, Z.; SU, J. Semantic neural machine translation using AMR. **Transactions of the Association for Computational Linguistics**, v. 7, p. 19–31, Mar. 2019. Citations on pages 19 and 21.

SONG, L.; ZHANG, Y.; WANG, Z.; GILDEA, D. A graph-to-sequence model for AMR-to-text generation. In: **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics**. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 1616–1626. Citation on page 19.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: **Advances in Neural Information Processing Systems**. Montreal, Canada: Curran Associates, Inc., 2014. p. 3104–3112. Citation on page 20.

UCHIDA, H.; ZHU, M.; SENTA, T. D. **UNL: Universal networking language–an electronic language for communication, understanding, and collaboration**. [S.l.], 1996. Citation on page 18.

VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer science & business media, 2013. Citation on page 136.

WANG, C.; XUE, N.; PRADHAN, S. A transition-based algorithm for AMR parsing. In: **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Denver, Colorado: Association for Computational Linguistics, 2015. p. 366–375. Citations on pages 21 and 22.

XUE, N.; CROFT, W.; HAJIC, J.; HUANG, C.-R.; OEPEN, S.; PALMER, M.; PUSTEJOVKSY, J. (Ed.). **Proceedings of the First International Workshop on Designing Meaning Representations**. Florence, Italy: Association for Computational Linguistics, 2019. Citation on page 21.

ZHANG, S.; MA, X.; DUH, K.; DURME, B. V. AMR parsing as sequence-to-graph transduction. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. Florence, Italy: Association for Computational Linguistics, 2019. p. 80–94. Citations on pages 21 and 136.

____. Broad-coverage semantic parsing as transduction. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 3784–3796. Citation on page 136.

ZHU, J.; LI, J.; ZHU, M.; QIAN, L.; ZHANG, M.; ZHOU, G. Modeling graph structure in transformer for better AMR-to-text generation. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 5458–5467. Citation on page 19.


ICMC
USP
SÃO CARLOS