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
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 (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 [11, 32, 24, 12, 33, 10], and, with the available parsers, some applications were developed and/or improved: automatic summarization [17], text generation [25, 26], entity linking [23, 7], 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 Section 3, we briefly introduce AMR fundamentals. Section 4 details our rule-based parser. In Section 5, we report the experiments and the obtained results. Finally, Section 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] considers 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. Fig. 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

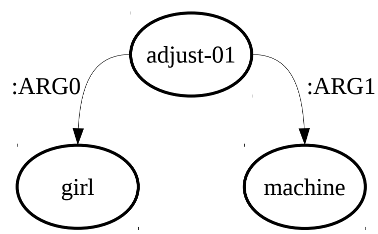

Fig. 3: AMR graph notation

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. Fig. 4 illustrates the similarity between the dependency tree (left) and the AMR graph (right).

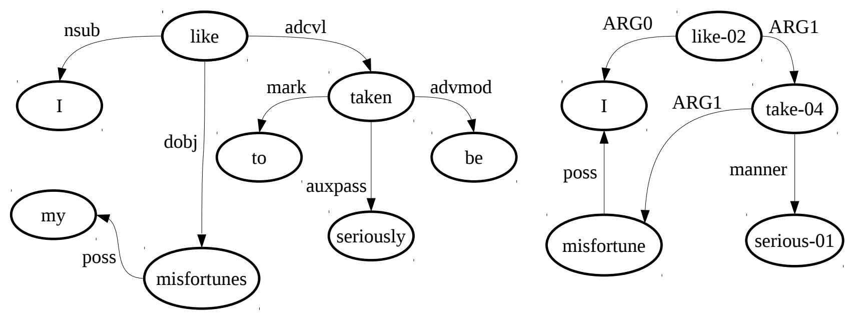

Fig. 4: Dependency tree and AMR graph for the sentence "I like my misfortunes to be taken seriously."

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. Fig. 5 shows the AMR graph for the sentence "At a glance I can distinguish China from

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

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 Figure 4).
- **:degree relation rule.** This rule creates a *:degree* relation when the parser produces a relation of adverbial modifier. Fig. 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".
- **:time relation rule.** This rule creates a *:time* relation when the SRL returns an *AM-TMP* argument. Fig. 9 shows an example for the sentence "The little prince said to me later on".

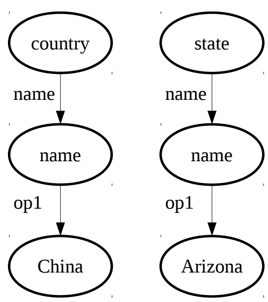

Fig. 5: Rule for named entity

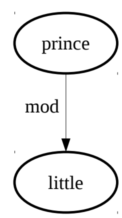

Fig. 6: Rule for :mod relation

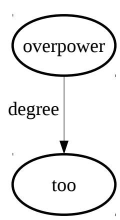

Fig. 7: Rule for :degree relation

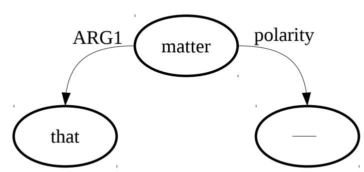

Fig. 8: Rule for negative polarity relation

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

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

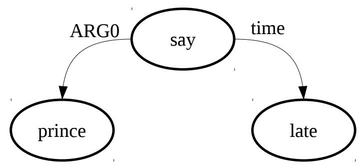

Fig. 9: Rule for :time relation

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.

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 the 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.

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

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].

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			

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., 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)
4. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus Universitetsforlag (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-Volume 1. pp. 581–589. Association for Computational Linguistics (2011)
6. Bos, J.: Expressive power of abstract meaning representations. Computational Linguistics pp. 527–535 (2016)
7. Burns, G.A., Hermjakob, U., Ambite, J.L.: Abstract meaning representations as linked data. In: International Semantic Web Conference. pp. 12–20 (2016)
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: International Conference on Computational Processing of the Portuguese Language. pp. 202–212 (2016)
14. Jurafsky, D., Martin, J.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall (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. (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 (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. Computational linguistics **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. Tokyo: UNU/IAS/UNL Center (1996)
29. Vanderwende, L.: Nlpwin–an introduction. Tech. rep., 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)