Procesamiento del Lenguaje Natural, Revista nº 68, marzo de 2022, pp. 85-97 recibido 19-12-2021 revisado 30-01-2022 aceptado 05-02-2022

# Low-resource AMR-to-Text Generation: A Study on Brazilian Portuguese

## *Generación de Texto a partir de AMR en Contexto de Bajos Recursos: Un Estudio para el Portugués Brasileño*

**Marco Antonio Sobrevilla Cabezudo, Thiago Alexandre Salgueiro Pardo**
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo
msobrevillac@usp.br, taspardo@icmc.usp.br

**Abstract:** This work presents a study of how varied strategies for tackling low-resource AMR-to-text generation for three approaches are helpful in Brazilian Portuguese. Specifically, we explore the helpfulness of additional *translated* corpus, different granularity levels in input representation, and three preprocessing steps. Results show that *translation* is useful. However, it must be used in each approach differently. In addition, finer-grained representations as characters and subwords improve the performance and reduce the bias on the development set, and preprocessing steps are helpful in different contexts, being delexicalisation and preordering the most important ones.
**Keywords:** AMR-to-Text Generation, Low-resource setting, Brazilian Portuguese.

**Resumen:** Este trabajo presenta un estudio de cómo diversas estrategias para abordar la generación de textos a partir de AMR en contextos de bajos recursos para tres enfoques son útiles en portugués brasileño. Específicamente, exploramos la utilidad de un corpus traducido, diferentes niveles de granularidad en la representación de entradas y tres técnicas de preprocesamiento. Los resultados muestran que el corpus traducido es útil. Sin embargo, debe usarse en cada enfoque de manera diferente. Además, las representaciones más detalladas, como las basadas en caracteres y subpalabras, mejoran el rendimiento y reducen el sesgo en el conjunto de validación, y los pasos de preprocesamiento son útiles en diferentes contextos, siendo la deslexicalización y el preordenamiento los más importantes.
**Palabras clave:** Generación de Texto a partir de AMR, Contexto de Bajos Recursos, Portugués Brasileño.

## 1 Introduction

Abstract Meaning Representation (AMR) is a semantic formalism that encodes the meaning of a sentence as a rooted, acyclic, labeled, and directed graph (Banarescu et al., 2013). This representation includes several semantic information, like semantic roles and named entities, among others.

AMR has become a relevant research topic in meaning representation, semantic parsing, and natural language generation (NLG). Its success is grounded on its attempt to abstract away from syntactic idiosyncrasies, and surface forms, its wide use of mature linguistic resources such as PropBank (Palmer, Gildea, and Kingsbury, 2005), and its usefulness on tasks like text summarisation (Liao, Lebanoff, and Liu, 2018), event detection (Li et al., 2015a) and machine translation (Song et al., 2019).

The goal of the AMR-to-Text generation task is to produce a text that represents the meaning encoded by an input AMR graph. For English, there are several works and approaches for this, as techniques of Statistical Machine Translation (Pourdamghani, Knight, and Hermjakob, 2016), tree and graph to string transducers (Flanigan et al., 2016) and, recently, neural models following sequence-to-sequence (Castro Ferreira et al., 2017; Konstas et al., 2017) and graph-to-sequence architectures (Beck, Haffari, and Cohn, 2018) or pretrained models (Mager et al., 2020; Ribeiro et al., 2020). For other languages, there are some multilingual work (Fan and Gardent, 2020) that tries to generate sentences in several languages. However, they use the AMR for English as input and do not capture some particular linguistic phenomena. In a different line, Sobrevilla Cabezudo, Mille, and Pardo (2019) try to generate Brazilian Portuguese (BP) sentences from the corresponding AMR for BP; nonetheless, the corpus is small (only 299 instances).

One problem that limits the research in other languages is the difficulty to get high-quality corpora (due to the difficult and expensive annotation task that it represents), resulting in smaller corpora and the inability for state-of-the-art methods to be replicated and/or achieve similar performance to the English ones.

It is well-known that the lack of data deteriorates the performance produced by neural models, which usually are data-hungry. To tackle this problem, some authors make use of data augmentation techniques, cross-lingual projection, and other strategies for increasing the corpus size (Hedderich et al., 2021). In the case of AMR-to-text generation, Sobrevilla Cabezudo, Mille, and Pardo (2019) proposed to translate both AMR and English sentences to their corresponding BP ones and then used the translated corpus as training/development set and a gold BP subset as test.

One problem associated with scarce corpus is data sparsity. Particularly, sparsity usually happens at input level in Natural Language Processing tasks. Word representation presents problems with unseen and rare words, resulting in low performance. Many works have proposed employing different granularities in input representation to solve this problem. The most commonly used are subwords (specifically Byte-pair encoding) (Sennrich, Haddow, and Birch, 2016) and characters, resulting in better results. In AMR-to-text generation, some work (Konstas et al., 2017; Mager et al., 2020) used finer-grained representations producing improvements; however, its benefits have not been studied in depth in low-resource settings.

This work explores three different strategies on three approaches for tackling low-resource AMR-to-text generation in Brazilian Portuguese. Specifically, we focus on machine translation and graph-to-sequence-based approaches and study the helpfulness of adding a *translated* corpus, using finer-grained representations and applying diverse preprocessing strategies.

It is worth noting that, even though the current state-of-the-art model for this task uses pretrained models (Mager et al., 2020; Ribeiro et al., 2020) and there are pretrained models for Brazilian Portuguese (Carmo et al., 2020), our goal is to show how to use simpler models and what kind of information could be helpful in low-resource settings or for other languages in which there are no pretrained models.

In general, our main contributions are:

- An analysis of the helpfulness of an additional translated corpus in different settings;
- An exploratory study about the effects of diverse granularity levels in input representation for low-resource AMR-to-text generation; and,
- A deep analysis of three commonly used preprocessing strategies in AMR-to-text generation: delexicalisation, compression, and linearisation.

We start by briefly reviewing AMR fundamentals (Section 2) and presenting the main related work (Section 3). Section 4 reports the techniques and methods that we investigate, while the achieved results are discussed in Section 5. Section 6 concludes this paper.

## 2 *Abstract Meaning Representation*

As previously mentioned, AMR aims to encode the meaning of a sentence in a directed, labeled, acyclic, and rooted graph (Banarescu et al., 2013). Furthermore, this representation may comprehend semantic information related to semantic roles, named entities, spatial-temporal information and co-references, among others.

Figure 1 presents an example of an AMR graph for the sentence “The boy destroyed the room”. It is worth noting that, as AMR abstracts away the syntactic information, multiple possible sentences can correspond to this graph. This way, another possible sentence that represents the graph could be “the destruction of the room by the boy”.

The current AMR-annotated corpus for English contains 59,255 instances[1]. For Non-English languages, there are some efforts to

[1]https://catalog.ldc.upenn.edu/LDC2020T02

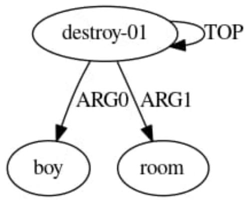


Figure 1: AMR example for the sentence "The boy destroyed the room.".

build corpora leveraging the alignments and existing parallel corpora by using AMR as an interlingua (Xue et al., 2014; Anchiêta and Pardo, 2018). Additionally, other works adapt the AMR guidelines to their languages (Sobrevilla Cabezudo and Pardo, 2019). However, most corpora are far from presenting a size similar to the English one.

For Brazilian Portuguese, as far as we know, there are two AMR corpora, one focused on annotating the sentences of "The Little Prince" book (Anchiêta and Pardo, 2018), and another one that contains manually annotated news text sentences (Sobrevilla Cabezudo and Pardo, 2019). Similarly to Banarescu et al. (2013), some concepts of both corpora were annotated using Verbo-Brasil (Duran and Aluísio, 2015), a lexical resource analogous to PropBank (Palmer, Gildea, and Kingsbury, 2005). Concerning the size of these corpora, the "Little Prince" corpus contains 1,527 annotated sentences (instances), and the second corpus comprises 299 instances, being both small and making it hard to replicate results obtained by state-of-the-art methods.

## *3 Related Work*

In the last years, several AMR-to-Text generation methods for English have been proposed. Initially, methods inspired on Statistical Machine Translation (SMT) techniques (Pourdamghani, Knight, and Hermjakob, 2016) and tree-to-string or graph-to-string transducers (Flanigan et al., 2016) were proposed. Recently, neural models as sequence-to-sequence (Neural Machine Translation or NMT) (Castro Ferreira et al., 2017; Konstas et al., 2017) and, mainly, graph-to-sequence (Beck, Haffari, and Cohn, 2018) and pretrained-based ones (Mager et al., 2020), have emerged, outperforming the previous approaches.

To the extent of our knowledge, the only work focused on AMR-to-Text generation for a Non-English language is proposed by Sobrevilla Cabezudo, Mille, and Pardo (2019). The authors explore the automatic construction of an AMR corpus for Brazilian Portuguese (BP) from its English version and evaluate SMT and NMT approaches on a BP test set composed of 299 instances. Other non-English work (Fan and Gardent, 2020) have tried to generate sentences in diverse languages from English AMR graphs. Although the results are promising, this work does not deal with some specific linguistic phenomena as the previous one does.

In what follows, we detail the dataset that we use in this work and the methods that we investigate.

## *4 AMR-to-Text Generation*

### 4.1 Data

The methods that we investigate are trained on two corpora and their combinations. The first one is an updated version of the AMR corpus for Brazilian Portuguese (Sobrevilla Cabezudo and Pardo, 2019), which represents our target (*gold*) dataset. This version is a manually annotated corpus comprising 870 instances divided into 402, 224, and 244 instances for training, development, and test, respectively.

The second one is a portion of an automatically generated AMR corpus for Portuguese and represents our augmented (*translated*) dataset. This corpus is generated by translating both AMR graphs and sentences from the English AMR corpus[2] to Portuguese and inheriting the alignments between node/edges and surface tokens[3] (Sobrevilla Cabezudo, Mille, and Pardo, 2019).

In general, this corpus comprises 18,219 and 1,027 instances in the training and development set, respectively, that correspond to the higher-quality translations according to BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007) scores.[4] It is worth noting that, differently from the work of Sobrevilla Cabezudo, Mille, and Pardo (2019), that translates only aligned concepts

[2]In this work, we use the LDC2016E25 corpus to perform the experiments.

[3]Surface tokens are those included in the reference sentence.

[4]The actual portion of the dataset contains 20,000 and 1,271 instances for training and development, respectively. However, some instances were filtered out because they presented some format errors.

in the AMR graphs, all concepts in the AMR graphs are translated.

## 4.2 Machine Translation-based Techniques

AMR-to-text generation receives an AMR graph as an input and generates a text in natural language; however, Machine Translation models are trained on linear input/output pairs. This way, we need to generate a flattened version of the AMR graph as input. Some flattened versions that have been used in the literature are the ones generated by the PENMAN notation (Matthiessen and Bateman, 1991) and the depth-first search (DFS) algorithm. However, other preprocessing steps can generate a flattened AMR version. Figure 2 shows an example of a flattened AMR version for the sentence *A crise na Venezuela foi um assunto que permeou as reuniões.* ("The crisis in Venezuela was an issue that permeated the meetings.").

In order to evaluate how the use of various flattened AMR versions affect the performance in AMR-to-text generation, we explore the strategies that include the preprocessing steps used by Castro Ferreira et al. (2017). In particular, the preprocessing steps are:

- Delexicalisation: that anonymises some entities of the graph;
- Compression: that determines which nodes and relations should be in the flattened graph; and,
- Linearisation: that determines how the nodes and relations should be put into the flattened graph.

We study two machine translation approaches, a statistical phrase-based one (Koehn, Och, and Marcu, 2003) as a strong baseline and one based on neural models (Bahdanau, Cho, and Bengio, 2015) in a similar way to Castro Ferreira et al. (2017).

### 4.2.1 Statistical Machine Translation (SMT)

The training parameters in SMT are the same of Castro Ferreira et al. (2017) and a 5-gram language model trained on the Brazilian Portuguese corpus provided by Hartmann et al. (2017) by using KenLM (Heafield, 2011). Furthermore, we use Moses (Koehn et al., 2007) to train the statistical machine translation models.

### 4.2.2 Neural Machine Translation (NMT)

The architecture and the parameters used in NMT are described as follows: the encoder and the decoder are a 1-layer RNN, and a 2-layers RNN with LSTM, each with a 512D hidden unit, respectively. Besides, the RNN decoder also uses bilinear attention (Luong, Pham, and Manning, 2015). Furthermore, the vocabulary is shared, and we apply weight tying between the source, target, and output layers. Additionally, source and target word embeddings are 512D each, and both are trained jointly with the model.

Among other parameters, the maximum sequence length in the decoder is 80, and we apply dropout with a probability of 0.25 in source embeddings. Moreover, models are trained using the Adam optimizer with a learning rate of 0.0003, a learning rate reduce factor of 0.5, and the learning rate decays if perplexity does not improve after 3 checkpoints/epochs. Besides, we use minibatches of size 16. Finally, we apply early stopping for model selection based on perplexity scores. Training is halted if a model does not improve on the development set for more than 8 checkpoints/epochs. Sockeye[5] (Hieber et al., 2017) provides all other parameters.

## 4.3 Graph-to-Sequence (G2S)

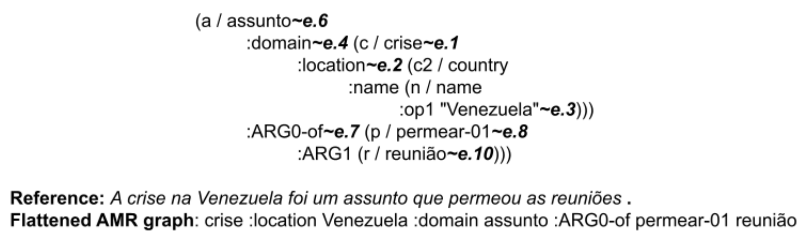
Unlike previous approaches, which depend on preprocessing steps and can lose information, the Graph-to-Sequence approach tries to capture the whole graph information more effectively. This work also follows the Graph-to-Sequence approach proposed by Beck, Haffari, and Cohn (2018), that models AMR graphs using a Gated Graph Neural Network (GGNN) (Li et al., 2015b).

In general, model input is defined by the nodes (concepts and relations) and positional embeddings of a graph. To consider AMR relations as nodes, the authors transform the original AMR graph into its respective Levi graph[6] (Levi, 1942). Finally, the output is a version of the original sentence.

We use the same architecture and parameters as Beck, Haffari, and Cohn (2018). Thus, the number of layers in the GGNN encoder

[5] `https://github.com/beckdaniel/sockeye/`

[6] A Levi graph is a modification of a labeled graph so that relations are converted into nodes generating an unlabeled graph.



```
(a / assunto~e.6
      :domain~e.4 (c / crise~e.1
            :location~e.2 (c2 / country
                  :name (n / name
                        :op1 "Venezuela"~e.3)))
      :ARG0-of~e.7 (p / permear-01~e.8
            :ARG1 (r / reunião~e.10)))
```

**Reference:** *A crise na Venezuela foi um assunto que permeou as reuniões .*
**Flattened AMR graph**: crise :location Venezuela :domain assunto :ARG0-of permear-01 reunião

Figure 2: Sentence *A crise na Venezuela foi um assunto que permeou as reuniões.* ("The crisis in Venezuela was an issue that permeated the meetings."), its corresponding AMR graph and a flattened version that includes only aligned nodes/edges. Alignments in AMR graph are in bold.

is 8. All dimensionalities are fixed at 512D except for the GGNN encoder, which uses 576D. The decoder uses a 2-layer LSTM and the Bilinear attention proposed by (Luong, Pham, and Manning, 2015). The remained parameters are the same as the NMT approach.

## 4.4 Preprocessing Strategies

The preprocessing strategies that we test in this work include:

- Delexicalisation: we delexicalise constants like named-entities or numbers, replacing the original information for tags such as __name1__ and __quant1__ for NMT (Castro Ferreira et al., 2017) and person_1 and quantity_1 for G2S (Beck, Haffari, and Cohn, 2018). A list of tag-values is kept, aiming to rebuild the output sentence after generation;
- Compression: it is performed using a Conditional Random Field (CRF) and executed sequentially over a flattened representation obtained by depth-first search through the AMR graph, and its name and the parent name represent each element. We use the CRF-Suite toolkit[7] (Okazaki, 2007) to train our model;
- Linearisation: we apply two strategies. The first consists of performing a depth-first search through the AMR graph, printing the elements (nodes and edges) according to the visiting order. The other strategy is based on the 2-step maximum entropy classifier developed by Lerner and Petrov (2013) and adapted by Castro Ferreira et al. (2017) (we called it preordering). Given an AMR graph represented by a tree, this consists of ordering a head and its corresponding subtrees, i.e., defining which subtrees should be at left/right of the head, and then ordering the subtrees in each built group (left and right side of the head).

All models are tested on inputs/outputs that include or not the preprocessing steps. However, we only explore compression and linearization (preordering) for SMT and delexicalisation for G2S. In addition, when compression is not considered, we include all elements from an AMR graph (nodes and edges).

## 4.5 Representation Levels

We explore three different representation levels for both input (AMR graph) and output (sentence): words, subwords, and characters. It is expected that finer-grained representations, such as subwords and characters, produce better results, handling in a better way rare words or even possible mismatches between the *translated* and the *gold* corpora.

Subwords are generated by using the Bertimbau's vocabulary provided by Souza, Nogueira, and Lotufo (2020)[8] that uses the sentencepiece tool[9] and the BPE algorithm (Sennrich, Haddow, and Birch, 2016). In the case of the flattened AMR graph, we do not decompose the relations. This way, relations such as "*:ARG0*" or "*:mod*" are kept intact, differently from concepts, such as "*ferida*", that are changed to "*fer ##ida*" in the case of subwords and "*f e r i d a*" in the case of characters.

It is worth mentioning that, in the case of G2S, each subword/character is represented

[7] `https://www.chokkan.org/software/crfsuite/`

[8] `https://github.com/neuralmind-ai/portuguese-bert`

[9] `https://github.com/google/sentencepiece`

by a node, and all subwords/characters that compose a concept are linked sequentially in two directions. For example, we create an edge from subword “fer” to “##ida” and vice-versa.

We present and analyze the achieved results in what follows.

## 5 *Results and Analysis*

Tables 1, 2, 3, and 4 show the overall results for SMT, NMT and G2S approaches in terms of BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and chrF++ (Popović, 2017) evaluation metrics [10]. The tables contain the results when the *translated* corpus (T), the *gold* corpus (G), a join of the training *translated* and *gold* corpora (T + G), and a join of the training/development *translated* and *gold* corpora (T + G Train/dev) are used. In addition, the results of using some preprocessing steps and representation levels are shown. Preprocessing steps are identified as +D (delexicalisation), +C (compression), and +P (preordering) and the opposite when these are not included in the preprocessing.

In general, the best result[11] for SMT happens when we train the model on T + G and use compression and preordering. Likewise, the best result for NMT occurs when the training is performed on T + G, using delexicalisation and preordering, and char-level representations. At last, G2S performs better when the model is trained on T + G train/dev, and lexicalisation and bpe-level presentation are applied.

Results on *gold* corpus show that SMT is by far the best approach to be used in the case of low-resource settings. It is expected as neural models usually need lots of data to achieve good performance, and SMT uses a pre-built language model that guides the decoding, differently from NMT and G2S in which the language model is built during training. In particular, using compressing (+C) and preordering (+P) produces the best results, being preordering the most critical preprocessing step, similarly to the results obtained by Castro Ferreira et al. (2017).

Concerning neural models, NMT produces the best performance; however, this is far from the SMT one yet. Char-level representation and Delexicalisation (+D) are the best strategies when BLEU is evaluated. However, lexicalisation (-D) is better when the metric is chrF++. Moreover, preordering (+P) seems useful when char-level representation is used. Finally, G2S presents the worst performance, being char-level representation and delexicalisation (+D) the best strategies.

In the following subsections, we will study how the performance changes in different contexts and try to answer three questions: (1) how helpful is the *translated* corpus? (2) what are the most useful preprocessing steps? (3) how fine-grained should be the representations to achieve better performance?

### 5.1 How helpful is the *translated* corpus?

To determine the helpfulness of the *translated* corpus, we study the performance when models are trained on T and T + G.

In general, the *translated* corpus is helpful as all models trained on it present better results than models trained on only *gold* corpus, however, there exists a mismatch between *translated* and *gold* corpora, as values for all measures in development set are quite higher than the obtained in test set (see results on *translated* corpus - T). This behavior can be generated by domain mismatch, in which the vocabulary is different even though both corpora are on news, or by structure mismatch between AMR graphs, since *translated* AMR graphs are English-biased and can introduce noise during training (as its size is bigger than the *gold* corpus).

Regarding the change in the performance when *gold* corpus is added to the *translated* one (T + G), SMT gets leveraging the data increase better. On the other hand, NMT performance presents a slight improvement when *gold* corpus is added. Finally, the G2S performance slightly drops in all cases and can suggest that there is a structural mismatch between the *translated* and gold AMR graphs, as this approach considers structural information, different from SMT or NMT, which use a flattened version with some nodes/edges included in it.

In order to evaluate how to deal with the possible mismatch, we add the *translated* development set (1,027 instances) to the *gold* one as well. Table 4 shows the result for each

[10]We execute 4 runs for each experiment and show the mean and standard deviation for NMT and G2S.

[11]Best results are highlighted in bold in Tables.

| | | DEV | | | TEST | | |
|---|---|---|---|---|---|---|---|
| | | BLEU | METEOR | chrF++ | BLEU | METEOR | chrF++ |
| Gold | +C+P | 11.58 | 0.31 | 0.48 | 10.00 | 0.30 | 0.48 |
| | +C-P | 11.36 | 0.29 | 0.47 | 7.95 | 0.26 | 0.46 |
| | -C+P | 6.06 | 0.24 | 0.43 | 6.05 | 0.24 | 0.43 |
| | -C-P | 7.31 | 0.24 | 0.44 | 4.89 | 0.22 | 0.43 |
| Translated | +C+P | 27.18 | 0.45 | 0.57 | 9.98 | 0.29 | 0.47 |
| | +C-P | 26.10 | 0.44 | 0.56 | 10.50 | 0.28 | 0.46 |
| | -C+P | 23.73 | 0.42 | 0.55 | 10.47 | 0.30 | 0.48 |
| | -C-P | 24.02 | 0.42 | 0.55 | 7.83 | 0.26 | 0.46 |
| Translated + Gold | +C+P | **18.67** | **0.38** | **0.52** | **14.83** | **0.33** | **0.49** |
| | +C-P | 17.75 | 0.37 | 0.51 | 11.96 | 0.32 | 0.47 |
| | -C+P | 17.38 | 0.37 | 0.51 | 13.91 | 0.32 | 0.49 |
| | -C-P | 14.86 | 0.35 | 0.50 | 11.96 | 0.32 | 0.48 |

Table 1: Overall SMT results.

setting and approach. Unlike the previous setting (T+G), both SMT and NMT present a small improvement in all metrics. However, G2S presents bigger improvements, suggesting that adding *translated* instances can make models more robust to possible structural divergences, leading to performance improvements.

## 5.2 What are the most useful preprocessing strategies?

### 5.2.1 Statistical Machine Translation

Pre-ordering (+P) seems to lead to improvements, however, this improvement is notorious when translated + *gold* corpora are used in the training set. Another point to highlight is the importance of compression (+C). Initial experiments (T and T + G) show that compression leads to slight improvements. However, no compression (-C) produces the best results when the classifier is trained on T + G train/dev.

### 5.2.2 Neural Machine Translation

Delexicalisation (+D) seems to be a good strategy for word and char-level representations, but it is not relevant for bpe-level. Moreover, compression (+C) generally harms the performance or produces mixed results, being better when lexicalisation (-D) is applied in char-level representation. Finally, pre-ordering (+P) seems to produce small improvements in all settings.

### 5.2.3 Graph-to-Sequence

About Graph-to-Sequence approach, Delexicalisation (+D) improves the performance when word and char-level presentations are used. However, the contrary happens when bpe-level representation is used. A possible explanation is that delexicalisation reduces data sparseness when word-level representation is applied together and allows to deal with large graphs in the case of char-level representation. However, in the case of bpe, delexicalisation seems to introduce noise and makes the model more prone to generate hallucinations.

## 5.3 How fine-grained should be the representations to achieve better performance?

Concerning the representation levels, characters and bpe produce the best and second-best performance for NMT. The main gain in both representations is in terms of METEOR and chRF++, which is expected as these representations are finer-grained and the evaluation measures take stems and characters into account.

Different from NMT, bpe produces the best performance for G2S. However, and as it was previously mentioned, this performance happens when delexicalisation is applied. This way, we hypothesise two possible problems: (1) word-level representations suffer more from mismatch problems as experiments on T and T + G show low performance, and (2) char-level representations can generate larger AMR graphs for which semantics can be challenging to be captured by G2S.

Another point to highlight is that finer-grained representations usually help reducing the bias to the development set, mainly when char-level representations are used. Consequently, mismatch problems are mitigated. This can be seen in the difference between development and test performance for experiments on T and T + G train/dev. For example, Figure 3 shows the difference mentioned for NMT. Experiments on T + G present a BLEU overall difference of 10.45, 9.9, and 5,67 between development and test for word, bpe, and char-level representations. Similarly, differences for METEOR and chrF++ are 0.11, 0.11, and 0.03, and 0.11, 0.09, and 0.00, respectively.

| | | | DEV | | | TEST | | |
|---|---|---|---|---|---|---|---|---|
| | | | BLEU | METEOR | chrF++ | BLEU | METEOR | chrF++ |
| G | word | +D+C+P | 0.00±0.00 | 0.06±0.00 | 0.05±0.00 | 2.66±0.14 | 0.10±0.00 | 0.13±0.01 |
| | | +D+C-P | 0.87±0.87 | 0.10±0.01 | 0.12±0.00 | 2.48±0.37 | 0.11±0.00 | 0.14±0.01 |
| | | +D-C+P | 0.00±0.00 | 0.10±0.01 | 0.11±0.00 | 2.61±0.23 | 0.11±0.00 | 0.13±0.00 |
| | | +D-C-P | 0.37±0.63 | 0.10±0.01 | 0.11±0.01 | 2.39±0.18 | 0.10±0.00 | 0.13±0.01 |
| | | -D+C+P | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 |
| | | -D+C-P | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 |
| | | -D-C+P | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 |
| | | -D-C-P | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 |
| | bpe | +D+C+P | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 | 0.00±0.00 | 0.01±0.00 | 0.01±0.00 |
| | | +D+C-P | 0.34±0.58 | 0.05±0.04 | 0.07±0.05 | 0.88±0.90 | 0.06±0.04 | 0.08±0.06 |
| | | +D-C+P | 0.33±0.56 | 0.07±0.03 | 0.09±0.04 | 1.33±0.81 | 0.08±0.04 | 0.10±0.05 |
| | | +D-C-P | 0.33±0.57 | 0.03±0.03 | 0.04±0.05 | 0.39±0.67 | 0.03±0.03 | 0.04±0.05 |
| | | -D+C+P | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 |
| | | -D+C-P | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 | 0.00±0.00 | 0.03±0.02 | 0.02±0.02 |
| | | -D-C+P | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 |
| | | -D-C-P | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 | 0.00±0.00 | 0.02±0.00 | 0.01±0.00 |
| | char | +D+C+P | 0.59±0.67 | 0.11±0.03 | 0.22±0.06 | 3.12±0.37 | 0.15±0.02 | 0.26±0.05 |
| | | +D+C-P | 1.61±1.00 | 0.11±0.01 | 0.19±0.01 | 2.80±0.27 | 0.11±0.00 | 0.19±0.00 |
| | | +D-C+P | 2.28±0.36 | 0.12±0.01 | 0.22±0.04 | 3.12±0.10 | 0.13±0.01 | 0.22±0.03 |
| | | +D-C-P | 1.63±0.09 | 0.10±0.00 | 0.18±0.00 | 2.88±0.35 | 0.11±0.00 | 0.19±0.00 |
| | | -D+C+P | 1.35±0.82 | 0.14±0.05 | 0.27±0.09 | 1.77±1.14 | 0.14±0.05 | 0.28±0.09 |
| | | -D+C-P | 0.00±0.00 | 0.11±0.00 | 0.26±0.00 | 0.48±0.82 | 0.13±0.01 | 0.27±0.01 |
| | | -D-C+P | 1.45±0.87 | 0.16±0.01 | 0.31±0.01 | 0.70±1.22 | 0.16±0.01 | 0.31±0.01 |
| | | -D-C-P | 0.72±0.74 | 0.09±0.04 | 0.20±0.07 | 0.00±0.00 | 0.09±0.04 | 0.19±0.07 |
| — | word | +D+C+P | 11.02±1.37 | 0.26±0.02 | 0.32±0.01 | 4.16±0.65 | 0.20±0.01 | 0.29±0.01 |
| | | +D+C-P | 4.66±0.19 | 0.18±0.01 | 0.24±0.01 | 2.46±0.29 | 0.13±0.00 | 0.19±0.00 |
| | | +D-C+P | 20.53±0.56 | 0.38±0.00 | 0.46±0.00 | 5.88±0.23 | 0.24±0.01 | 0.33±0.01 |
| | | +D-C-P | 19.35±0.92 | 0.37±0.01 | 0.44±0.00 | 5.88±0.30 | 0.23±0.00 | 0.32±0.01 |
| | | -D+C+P | 17.96±0.76 | 0.36±0.01 | 0.42±0.01 | 3.79±0.34 | 0.18±0.01 | 0.25±0.01 |
| | | -D+C-P | 2.32±0.37 | 0.12±0.01 | 0.16±0.01 | 0.12±0.21 | 0.06±0.00 | 0.09±0.01 |
| | | -D-C+P | 19.22±0.75 | 0.38±0.01 | 0.43±0.02 | 3.96±0.62 | 0.18±0.01 | 0.26±0.02 |
| | | -D-C-P | 19.81±0.77 | 0.37±0.01 | 0.42±0.01 | 3.17±0.33 | 0.17±0.01 | 0.24±0.01 |
| T | bpe | +D+C+P | 8.96±2.07 | 0.26±0.02 | 0.36±0.01 | 3.90±1.03 | 0.21±0.02 | 0.32±0.01 |
| | | +D+C-P | 12.89±3.52 | 0.33±0.02 | 0.44±0.02 | 3.57±1.10 | 0.21±0.02 | 0.33±0.01 |
| | | +D-C+P | 15.41±2.46 | 0.36±0.02 | 0.46±0.01 | 5.39±0.68 | 0.24±0.01 | 0.36±0.00 |
| | | +D-C-P | 20.04±0.60 | 0.38±0.00 | 0.48±0.01 | 7.05±1.00 | 0.27±0.02 | 0.38±0.01 |
| | | -D+C+P | 19.34±4.59 | 0.41±0.03 | 0.49±0.02 | 6.10±1.42 | 0.24±0.03 | 0.36±0.02 |
| | | -D+C-P | 13.60±2.37 | 0.36±0.02 | 0.46±0.01 | 2.86±0.74 | 0.19±0.01 | 0.32±0.01 |
| | | -D-C+P | 22.39±1.57 | 0.44±0.01 | 0.51±0.00 | 7.08±0.71 | 0.27±0.02 | 0.37±0.02 |
| | | -D-C-P | 20.87±1.16 | 0.42±0.01 | 0.50±0.01 | 5.47±0.63 | 0.24±0.01 | 0.35±0.00 |
| | char | +D+C+P | 13.39±0.37 | 0.27±0.00 | 0.37±0.00 | 8.69±1.33 | 0.29±0.01 | 0.43±0.01 |
| | | +D+C-P | 15.45±0.50 | 0.31±0.00 | 0.43±0.01 | 8.02±0.40 | 0.28±0.01 | 0.42±0.01 |
| | | +D-C+P | 13.73±0.40 | 0.31±0.00 | 0.43±0.01 | 8.21±0.95 | 0.28±0.01 | 0.42±0.01 |
| | | +D-C-P | 13.06±1.22 | 0.29±0.01 | 0.42±0.01 | 7.18±0.88 | 0.27±0.00 | 0.42±0.00 |
| | | -D+C+P | 16.06±2.91 | 0.33±0.04 | 0.43±0.03 | 7.63±2.23 | 0.28±0.03 | 0.42±0.03 |
| | | -D+C-P | 17.75±0.41 | 0.34±0.01 | 0.44±0.01 | 6.16±1.13 | 0.26±0.01 | 0.41±0.00 |
| | | -D-C+P | 15.73±1.19 | 0.33±0.02 | 0.43±0.02 | 6.97±1.40 | 0.26±0.02 | 0.41±0.02 |
| | | -D-C-P | 11.26±4.63 | 0.24±0.09 | 0.34±0.10 | 4.04±3.64 | 0.17±0.09 | 0.29±0.12 |
| T+G | word | +D+C+P | 2.77±0.57 | 0.16±0.01 | 0.22±0.02 | 4.76±0.38 | 0.20±0.01 | 0.28±0.02 |
| | | +D+C-P | 3.65±0.54 | 0.19±0.02 | 0.27±0.02 | 4.23±1.00 | 0.19±0.02 | 0.27±0.03 |
| | | +D-C+P | 5.15±0.82 | 0.23±0.01 | 0.31±0.01 | 6.04±0.30 | 0.22±0.01 | 0.30±0.01 |
| | | +D-C-P | 4.42±0.52 | 0.20±0.01 | 0.28±0.01 | 4.81±0.64 | 0.20±0.01 | 0.27±0.02 |
| | | -D+C+P | 2.93±0.73 | 0.17±0.01 | 0.24±0.00 | 3.59±0.38 | 0.18±0.00 | 0.24±0.00 |
| | | -D+C-P | 2.70±0.48 | 0.14±0.01 | 0.20±0.01 | 2.58±0.67 | 0.14±0.02 | 0.20±0.01 |
| | | -D-C+P | 3.51±0.77 | 0.16±0.02 | 0.23±0.02 | 2.57±0.27 | 0.16±0.02 | 0.22±0.02 |
| | | -D-C-P | 3.63±0.89 | 0.17±0.01 | 0.24±0.02 | 2.99±0.80 | 0.16±0.01 | 0.23±0.01 |
| | bpe | +D+C+P | 2.72±0.73 | 0.19±0.01 | 0.30±0.01 | 4.71±0.38 | 0.23±0.01 | 0.34±0.01 |
| | | +D+C-P | 3.38±1.35 | 0.20±0.04 | 0.32±0.03 | 3.21±1.43 | 0.19±0.04 | 0.31±0.03 |
| | | +D-C+P | 7.10±1.10 | 0.28±0.02 | 0.39±0.02 | 7.52±1.10 | 0.28±0.02 | 0.37±0.01 |
| | | +D-C-P | 5.68±1.21 | 0.26±0.02 | 0.37±0.02 | 5.78±1.38 | 0.25±0.02 | 0.35±0.01 |
| | | -D+C+P | 3.56±0.52 | 0.21±0.01 | 0.34±0.01 | 4.47±1.21 | 0.22±0.02 | 0.35±0.01 |
| | | -D+C-P | 4.45±1.02 | 0.22±0.02 | 0.33±0.01 | 4.60±1.36 | 0.22±0.02 | 0.34±0.02 |
| | | -D-C+P | 7.10±0.40 | 0.27±0.00 | 0.37±0.01 | 7.42±0.70 | 0.26±0.01 | 0.36±0.01 |
| | | -D-C-P | 6.69±0.77 | 0.26±0.01 | 0.36±0.01 | 5.93±1.35 | 0.25±0.01 | 0.36±0.01 |
| | char | +D+C+P | 7.82±0.44 | 0.26±0.01 | 0.38±0.01 | 9.38±0.22 | 0.30±0.01 | 0.44±0.01 |
| | | +D+C-P | 8.36±0.51 | 0.29±0.01 | 0.42±0.01 | 8.65±0.90 | 0.28±0.01 | 0.42±0.01 |
| | | **+D-C+P** | **7.28±0.49** | **0.29±0.01** | **0.42±0.01** | **10.03±0.37** | **0.31±0.01** | **0.44±0.01** |
| | | +D-C-P | 7.04±0.14 | 0.27±0.00 | 0.42±0.00 | 7.34±0.88 | 0.27±0.01 | 0.41±0.01 |
| | | -D+C+P | 7.48±0.74 | 0.29±0.01 | 0.43±0.01 | 8.85±0.78 | 0.29±0.01 | 0.43±0.01 |
| | | -D+C-P | 7.99±1.57 | 0.27±0.01 | 0.41±0.01 | 7.96±0.69 | 0.27±0.01 | 0.42±0.01 |
| | | -D-C+P | 5.98±0.59 | 0.27±0.02 | 0.41±0.02 | 8.25±0.94 | 0.29±0.02 | 0.43±0.02 |
| | | -D-C-P | 5.33±1.89 | 0.23±0.05 | 0.37±0.05 | 5.20±3.06 | 0.24±0.05 | 0.38±0.05 |

Table 2: Overall NMT results.

## 5.4 Manual Revision

We present now some analysis of actual generated cases. Figure 4 shows the AMR graph, the reference, and the output generated by the three approaches for the sentences "He/She does not want it" ("*não quer*") and "He/She attended excellent schools, and majored in economics at Yale." ("*frequentou excelentes escolas, e se formou em economia por Yale.*"). We can see some mistakes for each approach associated with hidden subjects (highlighted in red), wrong conjugation (blue), fluency/concordance (green), repetitions (purple), random words (yellow), and entity copying (pink).

The first example is simple, and the three approaches present similar outputs. SMT produces almost the same reference; however,

| | | | DEV | | | TEST | | |
|---|---|---|---|---|---|---|---|---|
| | | | BLEU | METEOR | chrF++ | BLEU | METEOR | chrF++ |
| G | word | +D | 0.00 ±0.00 | 0.03 ±0.01 | 0.02 ±0.01 | 0.00 ±0.00 | 0.03 ±0.01 | 0.02 ±0.01 |
| | | -D | 0.00 ±0.00 | 0.03 ±0.01 | 0.02 ±0.01 | 0.00 ±0.00 | 0.03 ±0.01 | 0.02 ±0.01 |
| | bpe | +D | 0.00 ±0.00 | 0.03 ±0.01 | 0.03 ±0.02 | 0.00 ±0.00 | 0.03 ±0.02 | 0.03 ±0.02 |
| | | -D | 0.00 ±0.00 | 0.02 ±0.01 | 0.01 ±0.00 | 0.00 ±0.00 | 0.02 ±0.01 | 0.01 ±0.00 |
| | char | +D | 0.00 ±0.00 | 0.09 ±0.01 | 0.13 ±0.01 | 1.59 ±0.47 | 0.09 ±0.01 | 0.14 ±0.01 |
| | | -D | 0.00 ±0.00 | 0.05 ±0.00 | 0.09 ±0.00 | 0.00 ±0.00 | 0.05 ±0.00 | 0.09 ±0.00 |
| T | word | +D | 14.88 ±4.17 | 0.32 ±0.06 | 0.38 ±0.06 | 4.66 ±1.50 | 0.18 ±0.04 | 0.26 ±0.05 |
| | | -D | 10.41 ±4.20 | 0.24 ±0.06 | 0.30 ±0.07 | 1.95 ±1.74 | 0.13 ±0.04 | 0.19 ±0.05 |
| | bpe | +D | 8.44 ±1.60 | 0.23 ±0.02 | 0.29 ±0.01 | 2.60 ±0.37 | 0.14 ±0.00 | 0.22 ±0.01 |
| | | -D | 21.04 ±1.09 | 0.42 ±0.01 | 0.48 ±0.00 | 6.75 ±0.51 | 0.26 ±0.01 | 0.36 ±0.01 |
| | char | +D | 11.46 ±1.67 | 0.25 ±0.02 | 0.32 ±0.03 | 6.07 ±2.02 | 0.23 ±0.04 | 0.35 ±0.05 |
| | | -D | 7.09 ±2.24 | 0.18 ±0.03 | 0.24 ±0.02 | 1.43 ±0.78 | 0.12 ±0.03 | 0.23 ±0.03 |
| T+G | word | +D | 3.52 ±2.14 | 0.17 ±0.05 | 0.23 ±0.05 | 3.80 ±2.01 | 0.16 ±0.04 | 0.23 ±0.05 |
| | | -D | 1.00 ±1.74 | 0.10 ±0.04 | 0.15 ±0.05 | 1.00 ±1.72 | 0.09 ±0.04 | 0.15 ±0.06 |
| | bpe | +D | 1.37 ±0.35 | 0.12 ±0.01 | 0.18 ±0.00 | 1.82 ±0.32 | 0.12 ±0.01 | 0.19 ±0.01 |
| | | -D | 5.62 ±0.43 | 0.26 ±0.01 | 0.36 ±0.01 | 6.44 ±0.79 | 0.26 ±0.01 | 0.36 ±0.01 |
| | char | +D | 5.21 ±1.25 | 0.22 ±0.03 | 0.33 ±0.05 | 6.09 ±1.50 | 0.22 ±0.04 | 0.34 ±0.05 |
| | | -D | 2.53 ±1.63 | 0.17 ±0.04 | 0.28 ±0.05 | 2.63 ±1.94 | 0.17 ±0.04 | 0.29 ±0.05 |

Table 3: Overall G2S results.

| | | | DEV | | | TEST | | |
|---|---|---|---|---|---|---|---|---|
| | | | BLEU | METEOR | chrF++ | BLEU | METEOR | chrF++ |
| SMT | word | +C+P | 25.66 | 0.43 | 0.56 | 12.92 | 0.31 | 0.48 |
| | | +C-P | 24.72 | 0.42 | 0.55 | 12.52 | 0.31 | 0.48 |
| | | -C+P | 22.09 | 0.41 | 0.54 | 14.69 | 0.34 | 0.50 |
| | | -C-P | 22.29 | 0.41 | 0.54 | 10.03 | 0.30 | 0.48 |
| NMT | word | +D+C+P | 11.21 ±1.36 | 0.25 ±0.01 | 0.32 ±0.02 | 5.38 ±1.03 | 0.22 ±0.02 | 0.30 ±0.02 |
| | | +D+C-P | 14.25 ±0.92 | 0.31 ±0.01 | 0.39 ±0.01 | 4.95 ±0.52 | 0.21 ±0.01 | 0.29 ±0.01 |
| | | +D-C+P | 16.82 ±0.81 | 0.34 ±0.01 | 0.42 ±0.00 | 6.70 ±0.79 | 0.24 ±0.01 | 0.32 ±0.01 |
| | | +D-C-P | 17.10 ±0.47 | 0.34 ±0.00 | 0.42 ±0.00 | 6.68 ±0.20 | 0.23 ±0.00 | 0.32 ±0.01 |
| | | -D+C+P | 14.88 ±1.42 | 0.32 ±0.01 | 0.38 ±0.01 | 3.94 ±0.64 | 0.19 ±0.01 | 0.26 ±0.01 |
| | | -D+C-P | 14.98 ±1.48 | 0.31 ±0.01 | 0.37 ±0.01 | 3.25 ±0.51 | 0.17 ±0.01 | 0.24 ±0.01 |
| | | -D-C+P | 17.64 ±0.74 | 0.35 ±0.01 | 0.41 ±0.01 | 4.76 ±0.44 | 0.20 ±0.01 | 0.28 ±0.01 |
| | | -D-C-P | 16.87 ±0.47 | 0.33 ±0.01 | 0.39 ±0.01 | 4.48 ±0.31 | 0.19 ±0.00 | 0.26 ±0.01 |
| | bpe | +D+C+P | 11.81 ±0.43 | 0.28 ±0.01 | 0.37 ±0.01 | 6.65 ±1.10 | 0.25 ±0.01 | 0.35 ±0.01 |
| | | +D+C-P | 14.32 ±0.87 | 0.33 ±0.01 | 0.43 ±0.01 | 5.09 ±0.54 | 0.24 ±0.01 | 0.35 ±0.01 |
| | | +D-C+P | 16.98 ±3.23 | 0.37 ±0.02 | 0.47 ±0.02 | 7.70 ±1.53 | 0.27 ±0.01 | 0.38 ±0.01 |
| | | +D-C-P | 16.32 ±2.56 | 0.36 ±0.02 | 0.45 ±0.01 | 6.15 ±0.87 | 0.26 ±0.01 | 0.36 ±0.01 |
| | | -D+C+P | 13.80 ±3.03 | 0.35 ±0.03 | 0.46 ±0.01 | 5.61 ±0.82 | 0.24 ±0.02 | 0.36 ±0.02 |
| | | -D+C-P | 14.53 ±3.18 | 0.35 ±0.02 | 0.45 ±0.01 | 4.79 ±1.55 | 0.22 ±0.02 | 0.34 ±0.02 |
| | | -D-C+P | 21.38 ±0.93 | 0.41 ±0.01 | 0.48 ±0.01 | 7.80 ±0.77 | 0.27 ±0.01 | 0.37 ±0.01 |
| | | -D-C-P | 20.25 ±1.06 | 0.40 ±0.01 | 0.49 ±0.01 | 6.38 ±1.16 | 0.26 ±0.01 | 0.38 ±0.01 |
| | char | +D+C+P | 12.61 ±0.50 | 0.27 ±0.00 | 0.37 ±0.00 | 9.42 ±0.47 | 0.30 ±0.00 | 0.44 ±0.00 |
| | | +D+C-P | 14.59 ±0.43 | 0.31 ±0.00 | 0.43 ±0.01 | 9.07 ±0.80 | 0.29 ±0.02 | 0.43 ±0.01 |
| | | +D-C+P | 13.20 ±0.16 | 0.31 ±0.00 | 0.43 ±0.01 | 9.83 ±0.88 | 0.30 ±0.01 | 0.44 ±0.01 |
| | | +D-C-P | 12.91 ±0.53 | 0.30 ±0.01 | 0.42 ±0.01 | 8.49 ±0.88 | 0.29 ±0.01 | 0.42 ±0.01 |
| | | -D+C+P | 17.18 ±0.54 | 0.35 ±0.00 | 0.45 ±0.00 | 10.14 ±0.38 | 0.30 ±0.01 | 0.44 ±0.01 |
| | | -D+C-P | 16.65 ±0.72 | 0.33 ±0.01 | 0.44 ±0.01 | 8.10 ±0.88 | 0.28 ±0.01 | 0.42 ±0.01 |
| | | -D-C+P | 12.19 ±4.38 | 0.27 ±0.08 | 0.37 ±0.09 | 5.93 ±3.48 | 0.24 ±0.10 | 0.36 ±0.12 |
| | | -D-C-P | 14.58 ±0.58 | 0.31 ±0.01 | 0.43 ±0.00 | 7.61 ±0.82 | 0.27 ±0.01 | 0.42 ±0.00 |
| G2S | word | +D | 16.84 ±1.88 | 0.36 ±0.02 | 0.43 ±0.02 | 7.70 ±1.74 | 0.26 ±0.03 | 0.34 ±0.03 |
| | | -D | 9.73 ±5.58 | 0.23 ±0.09 | 0.29 ±0.09 | 2.73 ±2.17 | 0.14 ±0.05 | 0.20 ±0.06 |
| | bpe | +D | 7.59 ±1.97 | 0.22 ±0.02 | 0.28 ±0.02 | 3.28 ±0.74 | 0.15 ±0.01 | 0.23 ±0.01 |
| | | **-D** | **20.85 ±1.21** | **0.41 ±0.02** | **0.48 ±0.02** | **8.69 ±0.59** | **0.29 ±0.02** | **0.38 ±0.02** |
| | char | +D | 11.10 ±1.95 | 0.25 ±0.03 | 0.32 ±0.02 | 7.03 ±2.46 | 0.24 ±0.04 | 0.35 ±0.05 |
| | | -D | 7.94 ±1.22 | 0.22 ±0.02 | 0.30 ±0.02 | 4.00 ±0.49 | 0.19 ±0.01 | 0.32 ±0.02 |

Table 4: Results of adding *translated* development set to the *gold* one. It is called T + G train/dev.

this includes the pronoun "*ele*" ("he/she") that is treated as a hidden subject in the reference. Conversely, NMT and G2S omit the pronoun, making the generated sentence more natural; nevertheless, both approaches generate the verb "*querer*" ("want") in a different conjugation (1st person). A possible explanation is that NMT and G2S are trained on char and bpe-level representations, this way, they can generate different conjugations easily. In addition, NMT generates the word "*dizer*" ("to say") that is not part of the AMR graph.

The second one is a harder example with more relations and concepts such as named entities ("university"), co-references ("e1 / *ele*" or "he/she") and connectors ("*e*"). In this case, none of the approaches can omit the pronoun "*ele*" as the reference does. Another common problem in all approaches is the lack of agreement/fluency. For example, the expression "na yale" should be replaced by "*em yale*" in order to be more fluent.

Analyzing other issues, SMT tries to generate sentences with all possible concepts included in the graph, even if the generated text is not fluent. On the other hand, neural models suffer from classical problems such as repetition and random word generation (the hallucination problem).

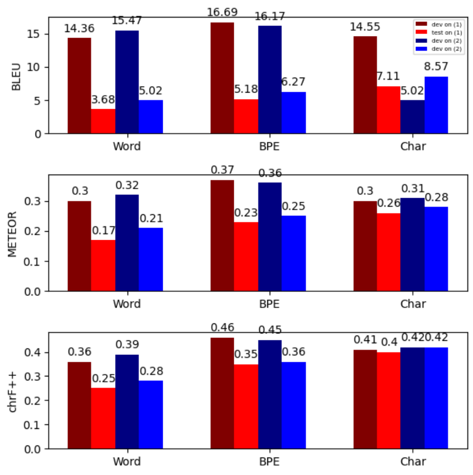


Figure 3: Difference between development and test performance for experiments on (1) T and (2) T + G train/dev.

```
(q / querer-01
    :polarity -
    :ARG0 (e / ele))
```

**Reference**: *não quer*
**SMT**: ***ele*** *não quer*
**NMT**: *não* ***quero dizer*** *.*
**G2S**: *não* ***quero*** *.*

```
(a / e
    :op1 (f / frequentar-01
        :ARG0 (e1 / ele)
        :ARG1 (e2 / escola
            :mod (e3 / excelente)))
    :op2 (f1 / formar-102
        :ARG1 e1
        :ARG2 (e4 / economia)
        :location (u / university
            :name (n / name
                :op1 "Yale"))))
```

**Reference**: *frequentou excelentes escolas , e se formou em economia por yale .*
**SMT**: ***ele*** *participou de uma excelente escola* ***formação*** *economia* ***na*** *yale*
**NMT**: ***ele*** *estava* ***formando excelente*** *e* ***formando*** *a economia* ***na*** *yale .* ***__name1__ :***
**G2S**: ***ele*** *estava* ***analisando*** *em excelente escola e foi* ***formada na*** *economia* ***na economia*** *.*

Figure 4: Outputs generated by the different approaches.

## 6 Conclusion and future work

This work presented a study of different strategies for tackling low-resource AMR-to-text generation for Brazilian Portuguese. We explore the helpfulness of additional translated corpus, different granularity levels in input representation, and three preprocessing strategies. It is worth noting this study can be helpful for work in other languages or meaning representations, mainly, when there is no pretrained models available.

Concerning the use of *translated* corpus, we can confirm its helpfulness. However, there are different contexts for each approach in which we can better leverage it. SMT improves its performance when the model is trained on the *translated* and *gold* corpora together. Neural models benefit from *translated* corpus more than SMT, even when these are trained on it solely. However, its join with the *gold* corpus can produce different results. In particular, G2S showed that there are structural divergences between *translated* and *gold* AMR graphs that can harm the performance when models are trained on both corpora. However, adding *translated* corpus to the development set allows to make the model more robust and achieve better performance.

About the representation levels, we highlight the use of finer-grained representations such as subwords and characters. Char-level seems to be the best option for NMT and bpe for G2S. However, it is worth noting that our study focuses on sentences of 23 tokens at maximum. This way, if we extend the work to longer sentences, bpe would probably performs better than char for NMT.

Finally, different combinations of preprocessing strategies are helpful for each approach, being preordering the best strategy for both machine translation approaches and delexicalisation for NMT. In the case of G2S, delexicalisation produces mixed results, being important just for word and char-level representations.

As future work, we plan to explore state-of-the-art approaches that are usually based on transformers, such as T5 (Ribeiro et al., 2020), or GPT-2 (Mager et al., 2020). Besides such issues, given some divergences between the *translated* and *gold* corpora that can harm the performance, it would be interesting to explore transfer learning for leveraging the knowledge learned from the *translated* corpus instead of training on both corpora together.

To the interested reader, more details about this work may be found at the web portal of the POeTiSA project at `https://sites.google.com/icmc.usp.br/poetisa`.

## Acknowledgments

The authors are grateful to CAPES and the Center for Artificial Intelligence (C4AI - `http://c4ai.inova.usp.br/`) of the University of São Paulo, sponsored by IBM and FAPESP (grant #2019/07665-4). Besides, this research has been carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0).

## References

Anchiêta, R. and T. Pardo. 2018. Towards AMR-BR: A SemBank for Brazilian Portuguese language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 974–979, Miyazaki, Japan. European Languages Resources Association.

Bahdanau, D., K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Beck, D., G. Haffari, and T. Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283. Association for Computational Linguistics.

Carmo, D., M. Piau, I. Campiotti, R. Nogueira, and R. Lotufo. 2020. Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. *arXiv preprint arXiv:2008.09144*.

Castro Ferreira, T., I. Calixto, S. Wubben, and E. Krahmer. 2017. Linguistic realisation as machine translation: Comparing different mt models for amr-to-text generation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 1–10, Santiago de Compostela, Spain. Association for Computational Linguistics.

Duran, M. S. and S. M. Aluísio. 2015. Automatic generation of a lexical resource to support semantic role labeling in Portuguese. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 216–221, Denver, Colorado. Association for Computational Linguistics.

Fan, A. and C. Gardent. 2020. Multilingual AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2889–2901, Online, November. Association for Computational Linguistics.

Flanigan, J., C. Dyer, N. A. Smith, and J. Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California, June. Association for Computational Linguistics.

Hartmann, N., E. Fonseca, C. Shulby, M. Treviso, J. Silva, and S. Aluísio. 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*, pages 122–131, Uberlândia, Brazil. Sociedade Brasileira de Computação.

Heafield, K. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

Hedderich, M. A., L. Lange, H. Adel, J. Strötgen, and D. Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online, June. Association for Computational Linguistics.

Hieber, F., T. Domhan, M. J. Denkowski, D. Vilar, A. Sokolov, A. Clifton, and M. Post. 2017. Sockeye: A toolkit for neural machine translation. *ArXiv*, abs/1712.05690.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens,

C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133. Association for Computational Linguistics.

Konstas, I., S. Iyer, M. Yatskar, Y. Choi, and L. Zettlemoyer. 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)*, pages 146–157, Vancouver, Canada, July. Association for Computational Linguistics.

Lavie, A. and A. Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the 2nd Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Lerner, U. and S. Petrov. 2013. Source-side classifier preordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 513–523, Seattle, Washington, USA, October. Association for Computational Linguistics.

Levi, F. W. 1942. Finite geometrical systems.

Li, X., T. H. Nguyen, K. Cao, and R. Grishman. 2015a. Improving event detection with abstract meaning representation. In *Proceedings of the First Workshop on Computing News Storylines*, pages 11–15, Beijing, China, July. Association for Computational Linguistics.

Li, Y., D. Tarlow, M. Brockschmidt, and R. Zemel. 2015b. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Liao, K., L. Lebanoff, and F. Liu. 2018. Abstract meaning representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Luong, T., H. Pham, and C. D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Mager, M., R. Fernandez Astudillo, T. Naseem, M. A. Sultan, Y.-S. Lee, R. Florian, and S. Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online, July. Association for Computational Linguistics.

Matthiessen, C. and J. A. Bateman. 1991. *Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese*. Pinter Publishers.

Okazaki, N. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

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

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Popović, M. 2017. chrF++: words helping character n-grams. In *Proceedings of* *the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark, September. Association for Computational Linguistics.

Pourdamghani, N., K. Knight, and U. Hermjakob. 2016. Generating English from abstract meaning representations. In *Proceedings of the 9th International Natural Language Generation conference*, pages 21–25, Edinburgh, UK, September 5-8. Association for Computational Linguistics.

Ribeiro, L. F. R., M. Schmitt, H. Schütze, and I. Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *CoRR*, abs/2007.08426.

Sennrich, R., B. Haddow, and A. Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Sobrevilla Cabezudo, M. A., S. Mille, and T. Pardo. 2019. 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 (MSR 2019)*, pages 94–103, Hong Kong, China, November. Association for Computational Linguistics.

Sobrevilla Cabezudo, M. A. and T. Pardo. 2019. Towards a general Abstract Meaning Representation corpus for Brazilian Portuguese. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 236–244, Florence, Italy, August. Association for Computational Linguistics.

Song, L., D. Gildea, Y. Zhang, Z. Wang, and J. Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Souza, F., R. Nogueira, and R. Lotufo. 2020. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23.*

Xue, N., O. Bojar, J. Hajič, M. Palmer, Z. Urešová, and X. Zhang. 2014. Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1765–1772, Reykjavik, Iceland, May. European Language Resources Association (ELRA).