SAPIENZA
UNIVERSITÀ DI ROMA

From Shallow to Whole-Sentence Semantics: Semantic Parsing in English and Beyond

Scuola di Dottorato in Informatica

Dottorato di Ricerca in Informatica – XXXIV Ciclo

Candidate

Rexhina Blloshmi
ID number 1740477

Thesis Advisor

Roberto Navigli

2021/2022

Thesis defended on 25 February 2022
in front of a Board of Examiners composed by:

Prof. Salvatore Gaglio (chairman)
Prof. Gabriella Pasi
Prof. Mauro Conti

External Reviewers:

Nathan Schneider
Shay Cohen

From Shallow to Whole-Sentence Semantics: Semantic Parsing in English and Beyond
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: rexhina.blloshmi@uniroma1.it

Dedikuar familjes sime

Wherever I go, I go with all my heart

Abstract

Humans want to speak to computers using the same language they speak to each other, rather than the symbolic and structured language machines are designed to process. Indeed, enabling a machine to process and interpret text automatically and then communicate verbally is one of the critical goals of the Natural Language Processing (NLP) and broader, the Artificial Intelligence (AI) fields. Moreover, computers are desired not to only process some written text, but also to understand it at the semantic and pragmatic level, which is further defined within the Natural Language Understanding (NLU) subfield. NLU aims at overcoming language ambiguities and complexities to enable machines to read and *comprehend* text. Therefore, to achieve this goal, we need computers capable of inputting text, preferably in any language, and parsing it into *semantic representations* which can be used as an interface between humans and computer language. To this end, a crucial issue faced by the NLP researchers is how to devise a language that is interpretable by machines and at the same time expresses the meaning of natural language, primarily known as the Semantic Parsing task. Semantic representations usually take the form of *graph-like structures* where words in a sentence are interconnected according to different semantic relations. Over time, this has garnered increasing attention, with researchers developing various formalisms that capture complementary aspects of meaning.

Two of the most popular formalisms in NLP that capture different levels of sentence semantics are Semantic Role Labeling (SRL) — often referred to as *shallow* Semantic Parsing — and Abstract Meaning Representation (AMR) — a popular *complete* formal language for Semantic Parsing — which includes SRL, among other NLP tasks. Both SRL and AMR have been widely studied in the NLP research, counting a large number of approaches to deal with task specificities and the challenges they pose, aiming at achieving human-like performance. In particular, the majority of the SRL works rely on task-specific sequence labeling approaches. In addition, they often make use of third-party components to solve subtasks of SRL, leading to non-end-to-end approaches. We observe a similar trend in AMR related research, where aspects of meaning are treated as a different constituent in a long pipeline. These complexities, which we will elaborate on more during this thesis, may hinder the effectiveness of the models in out-of-distribution settings while also making it more challenging to integrate SRL and AMR structures in downstream tasks of NLU efficiently. Another long-standing problem in NLP is that of enabling research in languages other than English. Especially in the context of AMR, the English dependency problem is even more evident provided that it was initially designed to represent the meaning of English sentences. In this thesis we investigate the aforementioned problems in SRL, including both dependency- and span-based SRL formulations, and in AMR, including AMR *parsing* — the task of converting utterances into an AMR graph — and its specular counterpart AMR *generation* — the task of generating natural language utterances from an AMR graph. We focus on relieving the burden of complex, task-specific architectures for English SRL and AMR casting them as sequence generation problems, motivated by the overgrowing success of general-purpose sequence-to-sequence methodologies in NLP in the recent years. Furthermore, we dispose of the previously necessary third-party dependencies in AMR *parsing*, thus achieving a full symmetry with its dual counterpart, AMR *generation*. Additionally, we make use of the sequence-to-sequence paradigm and transfer learning techniques to enable *cross-lingual* AMR parsing — the task of learning English-centric structures to represent meaning in multiple languages.

Acknowledgments

At the end of my PhD, which completes my cycle of studies, I look back and want to recognize the contribution of several people (of different nationalities) in my success.

<SQ> *Po i filloj falenderimet me familjen time, të cilës i dedikohet cdo sukses i imi. Falenderoj babin për besimin dhe suportin e përhershëm për të ndjekur rrugën e suksesit, brenda dhe jashtë Shqipërisë. Falenderoj mamin për suportin moral gjatë gjithë viteve të studimeve të mia, me bindjen që do ta kem po njësoj në sfidat që më presin. Falenderoj vëllain tim të madh, që nuk është lodhur kurrë duke shprehur sa krenar është për motrën e tij të vogël, PhD e parë të fisit. Faleminderit ba, ma, lali! Përfundoj këtë paragraph duke falenderuar shoqërine shqipëtare, Lea, Kejvi, Anxhi, Enxhi, dhe familjen time të madhe – jemi shumë – që më përkëdhelin kur më konsiderojnë si një shembull për tu ndjekur.* </SQ>

<IT> *Ringrazio il mio supervisore, tutti i miei amici-colleghi, e sopratutto i miei “amici magici", Caterina, Edoardo e Luigi, che hanno, senza dubbio, arricchito i miei anni di dottorato. Ringrazio Di Fabio per avermi “forzato" a parlare in italiano (anzi romano), anche se il mio italiano va migliorato.* </IT>

Finally I would like to thank Alexandre, for being a great support and for pushing me to give my best in the most stressful moments. <PT> *Obrigada!* </PT>

Contents

Chapter 1

Introduction

1.1 Overview

Human language bears vast complexities, especially related to how different constituents are interconnected in text. Consider the following example:

the fans desire Dua Lipa to meet them. (1.1)

When reading this piece of text, a human understands it refers to a group of people (*fans*) who want to meet with the celebrity they support (*Dua Lipa*). A machine instead, requires we put this information in a *structured* form that explicitly specifies the semantic relationships between a group of people and the act of desiring, a person and her role as a celebrity and finally, the group of people and the celebrity, i.e., protagonists in this phrase. Furthermore, while it is clear *them* refers to the fans in the above text, it is not easy for a computer to infer this relationship.

At the core of Natural Language Understanding (NLU) lies the task of Semantic Parsing, which aims at converting natural language utterances into an explicit machine-understandable semantic representation. Indeed, we arguably need these representations to make computers understand and, due to this, various formalisms have been developed for Semantic Parsing, based on different linguistic theories and covering distinctive levels of meaning aspects. Many broad-coverage meaning representations can be modeled as directed graphs, where nodes represent semantic concepts and directed edges represent semantic relations among them. However, language comprehension's difficulties caused the so-called semantic annotation balkanization; separate annotations exist for named entities, co-reference, semantic relations, discourse connectives, and temporal entities. In 2013, Abstract Meaning Representation (AMR) [Banarescu et al., 2013] emerged as a novel ambitious formalism aiming at being fully comprehensive, thus subsuming multiple traditional Natural Language Processing (NLP) tasks: Word Sense Disambiguation (WSD) [Bevilacqua et al., 2021b], Named Entity Recognition (NER) [Yadav and Bethard, 2018], Entity Linking (EL) [Ling et al., 2015], Coreference Resolution (CR) [Kobayashi and Ng, 2020], and finally, Semantic Role Labeling (SRL) [Màrquez et al., 2008] which in turn, is often entitled as *shallow* Semantic Parsing.

1.2 Meaning Representations

While SRL is not considered a representation, rather than a sequence tagging task that infers the predicate-argument structure of the sentence, the overlap of the latter with the popular AMR formalism makes it interesting for our study. SRL is commonly referred to as the task of automatically addressing the question "who did what, to whom, where, when, and how?" [Gildea and Jurafsky, 2002; Màrquez et al., 2008]. It is traditionally framed as either a dependency-based [Surdeanu et al., 2008a; Hajič et al., 2009] or a span-based [Carreras and Màrquez, 2005; Pradhan et al., 2012] sequence labeling task. Both dependency- and span-based SRL tasks consist in *four* conceptual components: i) predicate identification, ii) predicate disambiguation, iii) argument identification, and iv) argument classification/labeling. Their difference, instead, resides in the annotation used to represent the arguments, given a predicate in a sentence. The span-based SRL requires the identification and classification of the entire textual span of an argument. In contrast, dependency-based SRL is concerned about labeling only the head of the argument. However, the *primary* goal of the task is to determine the semantic relationship between a predicate and its arguments, while drawing these connections from a fixed set of arguments for the specific predicate, e.g., the PropBank [Palmer et al., 2005] inventory. In Figures 1.1 and 1.2 we show the respective dependency and span-based SRL annotations for *the fans desire Dua Lipa to meet them*. As

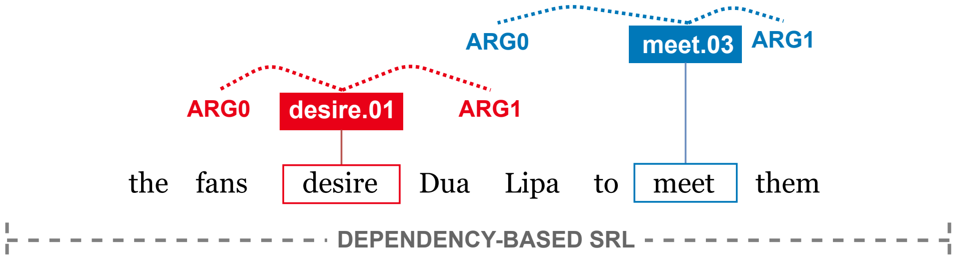


Figure 1.1. Dependency-based SRL annotations for the sentence *the fans desire Dua Lipa to meet them*.

one can see, both formulations share the same predicates, with equal senses and argument labels drawn from PropBank, with the only difference being the boundaries of the textual span for each argument. Even though researchers tend to agree that the two formalisms pose different challenges and capture complementary aspects of the overall task [Zhou et al., 2020a], there exist yet some aspects of meaning not covered in the example above by both SRL formulations, such as:

i) defining any relationship between various predicate-argument structures occurring in the same sentence;

ii) recognizing *Dua Lipa* as a named entity;

iii) linking *Dua Lipa* to the corresponding concept in an external knowledge base of entities; and finally,

iv) identifying the relationship between the *fans* and *them* in the phrase.

Furthermore, consider the following sentence:

$$\textit{the fans' desire is for Dua Lipa to meet them} \qquad (1.2)$$

While the meaning of this phrase is equivalent to that of the text in 1.1, the SRL annotations change substantially, despite the simple *syntactic* variation we applied, as shown in Figure 1.3. More specifically, the two variations differ in the i) set of predicates, i.e., `{be.01, meet.03}` $\not\subset$ `{desire.01, meet.03}`, ii) argument spans for the divergent predicates, and iii) argument labels of the unalike predicates. As a matter of fact, provided the *equivalent* meaning of the text in 1.1 and 1.2, it might be desirable to associate them to similar machine-

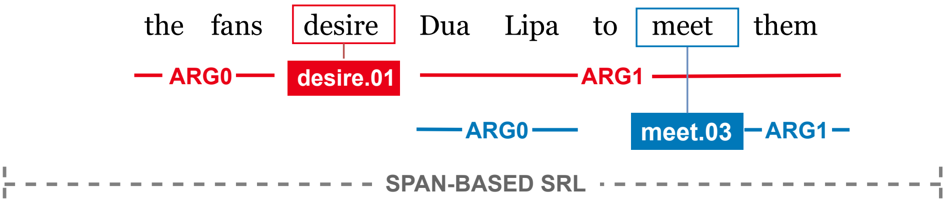


Figure 1.2. Span-based SRL annotations for the sentence *the fans desire Dua Lipa to meet them*.

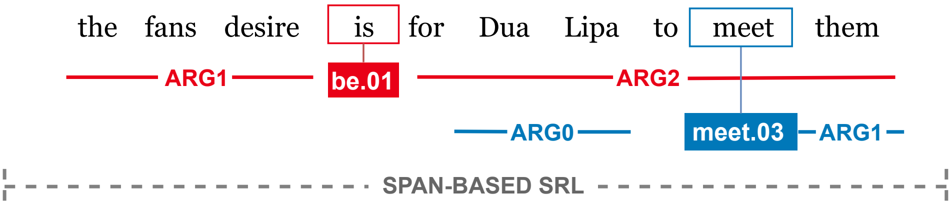


Figure 1.3. Span-based SRL annotations for the sentence *the fans desire is for Dua Lipa to meet them*.

readable structures. To this end, AMR handles all the aforelisted weaknesses of SRL, providing a complete representation of meaning.

AMR is a popular formalism for natural language that represents sentences as rooted, directed, and acyclic graphs, in which nodes are concepts and edges are semantic relations among them. AMR unifies, in a single structure, a rich set of information coming from different NLP tasks. Similarly to SRL, AMR draws its predicate-argument relations from the PropBank inventory. On the one hand, predicate-argument inventory is the main overlapping point of the two formalisms. On the other hand, differently from SRL, AMR is detached from the tokens in a text, thus abstracting away from syntactic variations. Indeed, this enables the association of equivalent sentences with the same semantic structure. In Figure 1.4[1] we show the AMR graph representing both variations of the example, i.e., *the fans desire Dua Lipa to meet them* and *the fans' desire is for Dua Lipa to meet them.*[2] This is in contrast with the SRL annotations in Figures 1.2 and 1.3, which show two different annotations for equivalent sentences. In addition to the meaning aspects captured by SRL, AMR also performs named entity recognition and entity linking, e.g., *Dua Lipa* is identified as a `PERSON` and is linked to the corresponding Wikipedia page.[3] Moreover, the node `fan`

[1] We use the style of SPRING demo to visualize the graphs: http://nlp.uniroma1.it/spring/

[2] *the fans want to meet Dua Lipa* is yet another simplified variation of the same sentence parsed with the same AMR graph.

[3] https://en.wikipedia.org/wiki/Dua_Lipa

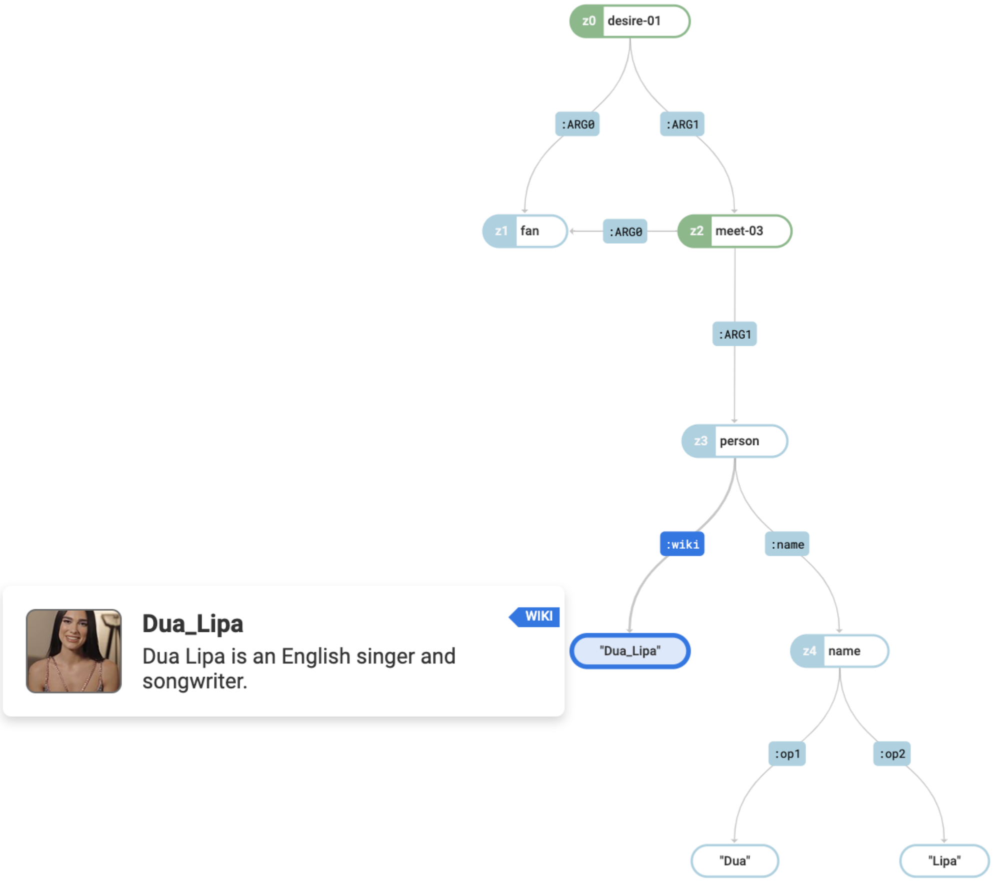


Figure 1.4. The AMR graph for the sentences *the fans desire Dua Lipa to meet them* and *the fans desire is for Dua Lipa to meet them.*

in Figure 1.4, plays a role in both predicates *desire.01* and *meet.03*, being the `Agent` and `Co-Agent`[4], respectively. The two incoming edges to the node `fan`, capture the coreference aspect, which is missing in the SRL annotations. Finally, AMR explicitly defines the relationship between multiple predicates in the sentence, e.g., the subgraph focused on `meet.03` plays the *thing wanted* role for the predicate `desire.01`.

In the last decade, both SRL and AMR have gained increasing attention in NLU research as two formalisms that capture aspects of meaning — overlapping, but yet at different levels — that could be beneficial for the ultimate goal of machine understanding. Numerous studies have found SRL to be beneficial in a wide range of downstream applications, not only in Natural Language Processing but also in Computer Vision, including: Question

[4] These human-readable labels are obtained from the VerbNet [Schuler, 2006] mappings available in PropBank.

Answering [Shen and Lapata, 2007], Machine Translation [Marcheggiani et al., 2018a], Visual Semantic Role Labeling [Gupta and Malik, 2015] and Situation Recognition [Yatskar et al., 2016]. Similarly, AMR's flexibility has resulted in promising improvements in Machine Translation [Song et al., 2019c], Text Summarization [Hardy and Vlachos, 2018; Liao et al., 2018], Human-Robot Interaction [Bonial et al., 2020a], Information Extraction [Rao et al., 2017] and, more recently, Question Answering [Lim et al., 2020; Bonial et al., 2020b; Kapanipathi et al., 2021]. However, since the meaning structures are automatically obtained using existing models, achieving human parity in SRL and AMR is regarded as a fundamental step towards NLU [Navigli, 2018] which can allow improvements of all abovelisted applications, *inter alia*.

1.3 Thesis Statement and Objectives

Thesis Statement. The overall goal of this dissertation is to develop computational approaches to structured predictions, viewing their learning from a different perspective than the majority of previous works, i.e., learning sequences rather than graph-like predictions. In particular, this thesis studies machine learning models to perform Semantic Role Labeling, AMR parsing — the task of converting a sentence into an AMR graph — and AMR generation — the specular task to AMR parsing. We focus on developing general-purpose and simple architectures that make minimal assumptions on the structure of the data and rely on transfer learning for performance enhancement across data domains and distributions. In addition, we propose the usage of different transfer learning techniques to pave the way towards better-performing AMR cross-lingual parsers for non-English sentences.

In what follows, we briefly overview the gaps in the literature, which we aim at overcoming and enumerate the objectives of this thesis.

1.3.1 Learning SRL and AMR as Sequences

Sequence-to-Sequence (seq-to-seq) learning was introduced as a general approach to sequence learning that makes minimal assumptions on the sequence structure [Sutskever et al., 2014]. While it was initially conceived for Machine Translation [Bahdanau et al., 2015], seq-to-seq learning rapidly found success in a variety of NLP tasks from Question Answering [Yin et al., 2016] to Dialogue [Song et al., 2019a], Text Generation [Lewis et al., 2020; Raffel et al., 2020]. While past and present studies have accomplished impressive results, the vast majority of the state-of-the-art models, proposed year after year, have framed SRL as a sequence labeling [Cai et al., 2018; Li et al., 2019; Conia and Navigli, 2020]. Indeed, only a tiny handful of studies have put forward SRL systems based on seq-to-seq learning, which fall behind traditional sequence labeling approaches in terms of performance [Daza and Frank, 2018]. Moreover, others can address only a portion of the SRL pipeline [Daza and Frank, 2019], making them an unappealing option for downstream applications. A similar trend has been observed in AMR research where predominant approaches to AMR parsing, i.e., the task of converting a sentence into an AMR graph, feature complex pipelines, in which the output of several different components is integrated [Zhang et al., 2019a,b; Cai and Lam, 2020a]. The AMR parsing performance of simpler, full seq-to-seq methods [Konstas et al., 2017; van Noord and Bos, 2017], has long lagged behind, mainly because they are less data-efficient than their alternatives. For learning SRL and AMR structures as sequences, we set the following objectives:

Objective 1. Devising ways to formulate and represent predicate-argument relations of SRL and AMR graphs as sequences, to enable seq-to-seq approaches generate graph-like sense and role annotations and AMR graphs, respectively, analyzing their positives and negatives.

Objective 2. Developing simple, versatile solutions to SRL which can achieve state-of-the-art results, previously attained only by sequence labeling approaches, for both dependency- and span-based English SRL.

Objective 3. Achieving symmetry in AMR parsing and generation, by designing a general-purpose seq-to-seq architecture for both directions and, therefore, reduce the complexity of AMR parsing model by disposing of the need of multi-step pipelines.

Objective 4. Attaining satisfying performance, not only in standard benchmarks for SRL and AMR, but also in different challenging settings which mimic out-of-distribution scenarios, thus allowing us to judge the generalizability of our approaches.

1.3.2 Enabling Cross-lingual AMR Parsing

A peculiar feature of the AMR formalism is that it aims at abstracting away from word forms. AMR graphs are *unanchored*, i.e., the linkage between tokens in a sentence and nodes in the corresponding graph is not explicitly annotated [Banarescu et al., 2013]. Hence, the feature of being agnostic about how to derive meanings from strings makes AMR particularly suitable for representing semantics cross-lingually. However, since AMR was initially designed for encoding the meaning of English sentences, the available resources and modeling techniques focus mainly on English while leaving cross-lingual AMR understudied [Damonte and Cohen, 2018]. For enabling cross-lingual AMR parsing, we set the following objectives:

Objective 5. Exploring different transfer learning techniques to enable learning AMR structures for non-English sentences despite the scarcity of cross-lingual training data.

Objective 6. Analyzing whether it is possible to transfer semantic structure information across different languages and whether or not AMR can be used to represent the meaning of sentences cross-lingually.

1.4 Thesis Contributions

In summary, the broad contributions of the dissertation to each objective are:

1. **GSRL** [Blloshmi et al., 2021b]: we present Generating Senses and RoLes (GSRL), the first end-to-end seq-to-seq model for Semantic Role Labeling. GSRL produces sequences of graph-like predicate-argument representations and attains state-of-the-art performance, which was previously achieved by sequence labeling approaches only. This work achieves Objective 2 and partly Objectives 1 and 4 and is elaborated in Chapter 3.

2. **SPRING** [Bevilacqua et al., 2021a; Blloshmi et al., 2021a]: we present Symmetric PaRsIng aNd Generation (SPRING), an end-to-end model for both AMR parsing and generation that relies on efficient graph linearization techniques and the expressive

power of a pretrained encoder-decoder to achieve unprecedented performance in both tasks [Bevilacqua et al., 2021a]. Then, we make SPRING available to the wide NLP community, but not only, through a highly interactive Web interface and RESTful APIs [Blloshmi et al., 2021a], thus paving the way to the integration of high-quality AMR structures in downstream tasks. These works accomplish Objective 3 and partly Objectives 1 and 4 and are detailed in Chapter 4.

3. **XL-AMR** [Blloshmi et al., 2020]: we present XL-AMR, a cross-lingual AMR parser that advances the state-of-the-art by a large margin, relying on transfer learning techniques to fill the gap of in-existent non-English training data and a seq-to-seq encoder to dispose of noisy word-to-node aligners. This work covers Objectives 5 and 6 and is explained in details in Chapter 5.

1.5 Publications and Personal Contributions

This dissertation is the result of a three-year research effort conducted mainly in the field of Semantic Parsing and partly in semantically-enhanced Information Retrieval. In what follows, we list the publications produced during these years chronologically, indicating those featured in this thesis. Each entry shows thorough referencing details, along with a brief description of individual contributions to each distinct work.

Included in this thesis:

1. **Rexhina Blloshmi**, Rocco Tripodi, and Roberto Navigli. *XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 2487–2500, November 2020.
   Personal Contributions: I have been the principal author and writer. I came up with the main idea, wrote the code, created several silver datasets included in the work, and planned and carried out the quantitative and qualitative experiments.

2. Michele Bevilacqua, **Rexhina Blloshmi**, and Roberto Navigli. *One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline*. In Proceedings of the 35th AAAI conference on Artificial Intelligence

(AAAI 2021), pages 12564-12573, February 2021.

Personal Contributions: I got into the project after the first author had started it, but my role was crucial for developing the work, and we achieved this publication as a partnership with the first author. I provided insights on the literature and state of the art, suggested and carried out the experiments (what to include, which dataset, etc.), as I was more knowledgeable in Semantic Parsing. I was fully involved in designing novel linearization techniques proposed, and exclusively in charge of the code for the quantitative and qualitative experiments regarding the usage of recategorization techniques. I was the primary writer for the Introduction, Related Work, and Analysis Sections, and contributed equally to Methodology and Experiments.

3. **Rexhina Blloshmi**, Simone Conia, Rocco Tripodi, Roberto Navigli. *Generating Senses and RoLes: An End-to-End Model for Dependency- and Span-based Semantic Role Labeling*. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), pages 3786-3793, August 2021.

   Personal Contributions: I have been the main author and writer (except for Introduction and Related Work). I came up with the main idea, wrote the code, and also planned and carried out the main experiments and, in part, the analytical experiments of the paper.

4. **Rexhina Blloshmi**, Michele Bevilacqua, Edoardo Fabiano, Valentina Caruso, and Roberto Navigli. *SPRING goes Online: End-to-End AMR Parsing and Generation*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021): System Demonstrations, pages 134–142, November 2021.

   Personal Contributions: I have been the main author and wrote the entire paper. The idea was developed in partnership with the second author (extension of our previous work). I did not contribute to the Web development, but I was actively suggesting the functionalities to include. Additionally, I carried out the main experiments of the paper.

Not included in this thesis:

5. **Rexhina Blloshmi**, Tommaso Pasini, Niccolò Campolungo, Somnath Banarjee, Roberto Navigli and Gabriella Pasi. *IR like a SIR: Sense-enhanced Information Retrieval for Multiple Languages*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021), pages 1030–1041, November 2021.

   Personal Contributions: I have been the main author and writer of the paper. I provided insights on the literature and state-of-the-art, suggested and carried out the experiments (what to include, which dataset, etc.) and the qualitative analysis.

6. Roberto Navigli, **Rexhina Blloshmi**, Abelardo Carlos Martìnez Lorenzo. *BMR: A Fully Semantic Meaning Representation to Overcome Language Barriers*. In Proceedings of the 36th AAAI conference on Artificial Intelligence (AAAI 2022): *Senior Member Track*, February 2022.

   Personal Contributions: While this is a Senior Member Track paper, the reason for my inclusion as a non-senior author is that the idea was developed jointly by the three authors and I also contributed in the writing of the paper.

7. Sveva Pepe, Edoardo Barba, **Rexhina Blloshmi**, Roberto Navigli. *STEPS: Semantic Typing of Event Processes with a Sequence-to-Sequence Approach*. In Proceedings of the 36th AAAI conference on Artificial Intelligence (AAAI 2022), February 2022.

   Personal Contributions: I co-supervised Sveva Pepe who developed this work for her master thesis. I contributed on the idea, experimental setup and wrote multiple sections of the paper.

∼ *This page was intentionally left blank* ∼

Chapter 2

Background and Related Work

2.1 Overview

In this Chapter, we mainly overview the two sentence-level representations we address in this dissertation, Semantic Role Labeling (SRL) and Abstract Meaning Representation (AMR), and research conducted about these formalisms in literature. In the center of both SRL and AMR lies the predicate-argument structure of a sentence and the question of “who did what, to whom, where, when, and how?”, crucial to enable text understanding. In addition, these formalisms make use of a common lexical resource of predicates, namely the Proposition Bank [Palmer et al., 2005, PropBank]. Nonetheless, while SRL addresses the shallow semantics of identifying the participants on an event, AMR also encompasses named entities, co-reference, negation, and modality.

In what follows, we briefly outline the existing semantic representations in the literature (Section 2.2) and then focus on SRL and AMR, describing the lexical resources used by both formalisms (Section 2.3), and then detailing the characteristics of SRL (Section 2.4) and AMR (Section 2.5) alongside the main approaches tackling them.

2.2 Broad-coverage Semantic Parsing

Semantic Parsing is defined as “the task of mapping natural language sentences into *complete formal meaning representations* which a computer can execute for some domain-specific application” [Kate and Wong, 2010]. Meaning representations are often based on an underlying formalism or grammar, on which machines can act. These formalisms can take the form of first-order logic and *lambda calculus* [Artzi et al., 2014], programming languages such as Python, SQL – also known as *executable* semantic parsing – and graph-based formalisms, namely broad-coverage semantic parsing. The latter has been judged as advantageous when compared to other formalisms, as they: i) are accessible for a human to read and interpret, and ii) are widely studied in the literature such that rich graph algorithms can be used for learning [Kamath and Das, 2019]. Furthermore, graph formalisms aim at encoding text in an abstract form that captures aspects of meaning that can be reusable in various scenarios, thus being *domain independent*. Examples of graph-based representations include Elementary Dependency Structures [Oepen and Lønning, 2006, EDS], Prague Tectogrammatical Graphs [Hajič et al., 2012, PTG], Universal Conceptual Cognitive Annotation [Abend and Rappoport, 2013, UCCA], Abstract Meaning Representation [Banarescu et al., 2013], Universal Decompositional Semantics [White et al., 2016, UDS], *inter alia*.

Most of these formalisms have been initially developed for representing English sentences only. Recently, various attempts have been made towards formally representing non-English texts as well. In this line of research, Abend and Rappoport [2013] proposed UCCA as a cross-lingual annotation that connects words in a sentence using semantic relations that are not language-specific. PTG [Hajič et al., 2012] is another formalism that enriches syntactic structures with the core predicate-argument relations of a sentence. Similar to AMR, PTG relies upon PropBank-like predicate inventories to represent non-English sentences. More recently, Abzianidze et al. [2017, PMB] propose a parallel meaning bank based on the Discourse Representation Theory, i.e., a formal logic meaning representation which includes syntactic and semantic annotations of sentences. PMB obtains non-English sentence representation by automatically projecting through English using one-to-one word alignments. In this thesis, we focus on the AMR formalism, focusing both in English and its cross-lingual applicability. We detail AMR later in this Chapter (Section 2.5).

2.3 The Proposition Bank

The Proposition Bank, commonly referred to as PropBank, is a large project that aims enabling the development of better language understanding systems [Palmer et al., 2005]. It provides an additional layer of predicate-argument information, or semantic role labels, to the syntactic structures of the Penn Treebank [Marcus et al., 1993]. As such, Propbank provides with a large inventory of English verbs associated with distinct senses, namely predicates, and a set of underlying semantic roles for each verb. In particular, it comprises of 10,687 framesets[1], with 6 different core role labels and 19 modifier roles. Due to the difficulty of defining a cross-frame set of thematic roles, PropBank annotates each verb sense with a specific set of enumerative roles such as {ARG0, ARG1, ARG2}, which are semantically defined within its frameset, and are mapped to human-readable labels from VerbNet [Schuler, 2006], e.g., {AGENT, PATIENT, THEME, EXPERIENCER}. Nevertheless, an important goal is to provide consistent argument labels across different syntactic realizations of the same verb. For instance, in both sentences below the *hearer* (ARG2) and the *utterance* (ARG1) for the predicate *tell* are assigned the same argument roles independent of the syntactical alterations, i.e., in the first sentence there exists a subject that acts upon the verb (*speaker* ARG0), while in the second sentence we have a passive voice.

[$_{\text{ARG0}}$The doctor] told [$_{\text{ARG2}}$the patient] [$_{\text{ARG1}}$to take the medicine]

[$_{\text{ARG2}}$ The patient] was told [$_{\text{ARG1}}$to take the medicine]

Below we display an excerpt of the PropBank framesets for the first sense [2] of the verbs *take* and *tell*:

take.01: *take, acquire, come to have, choose, bring with you from somewhere, internalize*

ARG0: *taker* (AGENT)

ARG1: *thing taken* (THEME)

ARG2: *taken FROM, SOURCE of thing taken* (SOURCE)

ARG3: *destination* (DESTINATION)

Example: [$_{\text{ARG0}}$She] **took** [$_{\text{ARG1}}$the law] [$_{\text{ARG3}}$into her own hands]

[1] A frameset corresponds to a verb sense which has a specific set of semantic arguments as per PropBank annotation guidelines (https://verbs.colorado.edu/~mpalmer/projects/ace/PBguidelines.pdf).

[2] Both predicate.xx (in SRL) and predicate-xx (in AMR) are used as equivalent notations to indicate the sense number of the predicate.

tell.01: *pass along information*

ARG0: *speaker* (AGENT)

ARG1: *utterance* (TOPIC)

ARG2: *hearer* (RECIPIENT)

Example: [$_{\text{ARG0}}$The scores] **tell** [$_{\text{ARG2}}$you] [$_{\text{ARG1}}$what the characters are thinking and feeling].

In these examples we only noticed numbered roles, which are part of the core argument roles set {ARG0, ARG1, ARG2, ARG3, ARG4, ARG5}. In addition to that, PropBank has a number of non-numbered arguments such as {ARG-TMP,ARG-LOC, ARG-DIR, ARG-MNR, ARG-CAU}, which represent verb modifiers addressing the questions “when, where, where to/from, how, why".

Similarly to PropBank, several inventories have been developed for languages other than English, such as Chinese [Xue and Palmer, 2003], Arabic [Palmer et al., 2008], Spanish and Catalan [Taulé et al., 2008], Hindi-Urdu [Bhatt et al., 2009], Basque [Aldezabal et al., 2010], Brazilian Portuguese [Duran and Aluísio, 2011], Finnish [Haverinen et al., 2015], and Turkish [Şahin and Adalı, 2018]. However, these inventories are language specific, differ in the set of roles used and are not linked among them. Moreover, this implies that a considerable amount of work will be needed for the creation of a corresponding resource for each new language of interest.

Nonetheless, the release of the PropBank corpus sparked a notable interest in SRL among researchers. Likewise, PropBank have been extensively used within the AMR formalism, which adapts and extends PropBank frames for abstracting away from syntactic idiosyncrasies.[3]

2.4 Semantic Role Labeling

Semantic Role Labeling is the sentence-level semantic analysis of text concerned with understanding the relations among an event, represented by a predicate, and its participants and properties in a sentence. Indeed, the predicate – usually a verb constituent – determines “what" happened and the other constituents express the “who”, “whom,” “where,” “when,”

[3] We will see later in this Chapter that AMR makes use of verbal frames to represent not only the verbs in a sentence but also other parts of speech, e.g., nouns, adjectives, whenever possible.

and "how" phenomena relevant to a certain predicate. These relations are drawn from a predefined inventory of verbs associated with their possible semantic roles such as PropBank, which we employ in our work.

SRL comprises of four conceptual components commonly defined as:

i) *predicate identification* which consists in detecting the predicates that express an event or convey an action;

ii) *predicate disambiguation* which is related to assigning an appropriate sense to each predicate drawn from a predefined inventory;

iii) *argument identification* which identifies the sentential constituents, called arguments, that participate in the event or action outlined by each predicate;

iv) *argument classification* that chooses the most appropriate relation, called semantic role, that governs each predicate-argument pair.

SRL is traditionally framed as either a dependency-based [Surdeanu et al., 2008a; Hajič et al., 2009] or a span-based [Carreras and Màrquez, 2005; Pradhan et al., 2012] labeling task. Given a predicate in a sentence, the difference between the two settings is in the formalism used to represent its arguments, where dependency-based SRL is concerned about identifying and classifying only the syntactic head of an argument, while span-based SRL identifies and classifies the whole textual span related to the argument.

Even if, to date, it is not clear whether one is better than the other [Li et al., 2019], researchers tend to agree that these two formalisms pose different challenges and capture complementary aspects of the overall task [Zhou et al., 2020a]. The first verb inventory used for SRL is FrameNet [Baker et al., 1998], which in turn, is based on frame semantics. In addition, there exist several SRL shared tasks which mainly derive their data from PropBank, such as CoNLL-2004 [Carreras and Màrquez, 2004], CoNLL-2005 [Carreras and Màrquez, 2005], and CoNLL-2012 [Pradhan et al., 2012] for span-based SRL, and CoNLL-2008 [Surdeanu et al., 2008b] and CoNLL-2009 [Hajič et al., 2009] for dependency-based SRL. The existence of these benchmarks allows advancements in approaches concerning both formulations. In this thesis we focus on the most recent benchmarks for dependency- and span-based SRL, CoNLL-2009 and CoNLL-2012, respectively, based on the PropBank verb inventory.

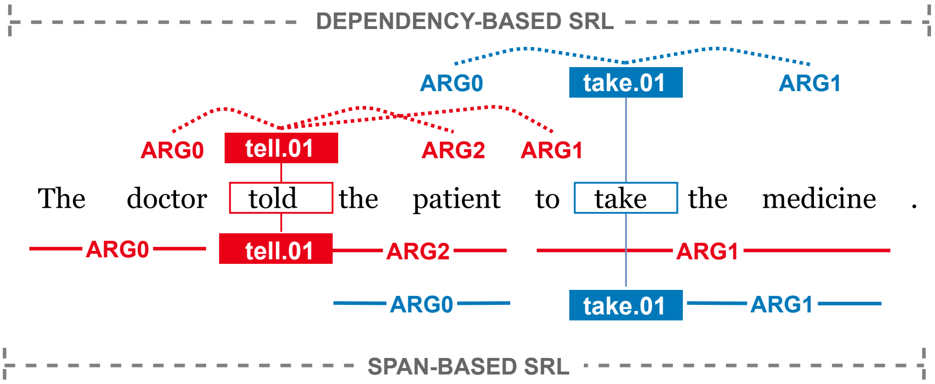


Figure 2.1. Example of a sentence with two predicates: dependency-based SRL (upper part) and span-based SRL (lower part).

In Figure 2.1 we illustrate an example annotated according to each formalism; span-based SRL which requires the identification and classification of the entire textual span of an argument, and dependency-based SRL which, instead, is concerned about labeling only the head of the argument. This sentence features two predicates, *told* and *take*, which are annotated with their senses according to the context, `tell.01` and `take.01` (their frames are displayed earlier in Section 2.3). If we consider the `take.01` predicate, *patient* is the *taker*, labeled as `ARG0` and *medicine* is the *thing taken* labeled as `ARG1`. According to its frameset (see Section 2.3), `take.01` might assume other semantic roles as well, such as *source* and *destination*. However, not all the roles defined within the frame are required to appear in a sentence. On the contrary, a predicate cannot assume any role that is not included in the Propbank frameset for the predicate. Each predicate-argument structure takes the form of a graph, with nodes being the predicate and the constituents of the sentence playing a role in this event, while the edges are semantic roles between them. This graph-like representation is in itself not necessary in SRL, which is often regarded as a sequence tagging task, but it is, in fact, beneficial in downstream application. Moreover, the same predicate-argument graph-like structure is employed for AMR, which is a graph-based formalism in itself.

2.4.1 Approaches to SRL

The earliest feature-based algorithms for SRL begin by parsing the input sentence to a parse tree using a broad-coverage parser. Then, using several tree traversals, first the predicates

are identified, then the nodes which might have a role for each predicate, and finally, through a supervised algorithm, these nodes are labeled with semantic role labels. Therefore, to simplify the task complexity, it was common to break it down into multiple steps rather than a single-stage classifier. However, these classifiers used to follow the simplifying assumption, such as predicates and their arguments can be labeled independently. Indeed, this is a false assumption as the label assignment for the arguments is global and their interactions matter according to the PropBank guidelines, e.g., a predicate cannot assume more than one argument of a specific role. Over the years, researchers made a great many steps forward in the design of better SRL models, moving from manually-engineered feature templates to multilayered neural networks [Cai et al., 2018; Marcheggiani and Titov, 2020], and from static to dynamically-contextualized word representations [He et al., 2019; Conia and Navigli, 2020], from English to other languages [Conia and Navigli, 2020; Conia et al., 2021]. As a matter of fact, the standard neural formulation of SRL is based on the IOB format (inside, outside, beginning) format [Ramshaw and Marcus, 1995], which is often used for the sequence tagging tasks.
While past and present studies have accomplished impressive results, the vast majority of the state-of-the-art models proposed year after year have framed SRL as a sequence labeling task [Cai et al., 2018; Li et al., 2019], and only a small handful of studies have put forward SRL systems based on seq-to-seq learning [Sutskever et al., 2014], despite the growing success of this paradigm in other areas of NLU [Yin et al., 2016; Lewis et al., 2020; Raffel et al., 2020].

Sequence-to-Sequence SRL. Despite the advancements in seq-to-seq learning, recent works in SRL predominantly revolve around sequence labeling approaches [Cai and Lapata, 2019b; Xia et al., 2019; Conia and Navigli, 2020; Marcheggiani and Titov, 2020; Conia et al., 2021], with no many attempts that formulate and tackle the task in a seq-to-seq fashion. In a potential seq-to-seq formulation, a model is tasked to maximize the conditional probability of a sequence comprising the predicate senses and semantic roles. Indeed, Daza and Frank [2018] and Daza and Frank [2019] are, to the best of our knowledge, the most notable studies on generation-based models for SRL. However, these SRL seq-to-seq models fall behind traditional sequence labeling approaches in terms of performance [Daza and Frank, 2018] and can address only a portion of the SRL pipeline [Daza and Frank, 2019], making them an unappealing option for downstream applications.

End-to-End SRL. Due to its complexity, SRL is often divided into a pipeline of four stages or subtasks handling predicate/argument identification and classification steps separately. While early work tried to develop distinct systems for each subtask, later studies successfully demonstrated that sequence labeling models [Cai et al., 2018; Li et al., 2019] can benefit from tackling some of these tasks jointly with multitask learning [Caruana, 1997]. However, seq-to-seq models proposed over the last few years can only solve the later stages of the SRL pipeline – namely, argument identification and argument classification – and, therefore, they still require an underlying system to perform at least predicate sense disambiguation [Daza and Frank, 2018, 2019]. Indeed, the function of a semantic role is often well-defined only with respect to a given predicate sense, especially when dealing with PropBank-like predicate-argument structure inventories. For example, even though there are two `ARG1` role labels in Figure 2.1, they actually encode different relations: when `ARG1` is associated with the predicate sense `tell.01`, it refers to the *utterance* or *topic* of the action, whereas, when it is an argument for the predicate sense `take.01`, it refers to the *thing taken* or *theme* of the action. While predicate sense disambiguation is essential to SRL, introducing structured predicate-argument relations in a seq-to-seq model is not trivial.

In summary, inspired by recent advances in seq-to-seq paradigm and innovative decoder-side pretraining [Lewis et al., 2020], in our work [Blloshmi et al., 2021b] we show that a *seq-to-seq* model is able to challenge *sequence labeling* systems across multiple gold benchmarks, in standard and synthetic evaluation settings. We explore different predicate-argument linearization schemes and introduce, to the best of our knowledge, the first end-to-end seq-to-seq model to successfully generate both sense and role labels (Chapter 3).

2.5 Abstract Meaning Representation

Abstract Meaning Representation [Banarescu et al., 2013] is a popular formalism for representing the semantics of natural language in a readable and hierarchical way. While it does not have an underlying theoretical formalism, AMR follows a neo-Davidsonian event

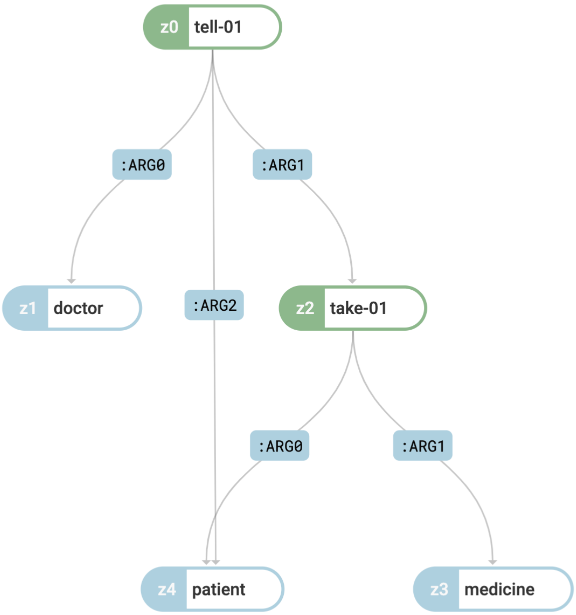


Figure 2.2. The AMR graph for the sentence *The doctor told the patient to take the medicine*.

specification [Davidson, 1969]. AMR pairs English sentences with graph-based logical formulas which are easily accessible by both humans and machines, while abstracting away from many syntactic variations. AMR encodes information about the predicate-argument structure, named entities and entity linking, coreference, polarity, and modality, *inter alia*. In Figure 2.2 we show the AMR parse for the sentence:

The doctor told the patient to take the medicine. (2.1)

Even though it is modeled as a graph, an AMR can be compactly represented and visualized using the PENMAN notation [Kasper, 1989; Goodman, 2020][4], i.e., the encoding that is used in the release files of AMR:

```
(z0 / tell-01
    :ARG0 (z1 / doctor)
    :ARG1 (z2 / take-01
              :ARG0 (z4 / patient)
              :ARG1 (z3))
    :ARG2 z4)
```

[4] We will use PENMAN notation throughout this thesis.

Additionally, an AMR can be represented as logical formulas composed of triples: These triples are mostly used for programmatically comparing the AMR graphs.

Root(z0, z0) $\wedge$
instance(z0, tell-01) $\wedge$
instance(z1, doctor) $\wedge$
instance(z2, take-01) $\wedge$
instance(z3, medicine) $\wedge$
instance(z4, patient) $\wedge$
ARG0(z0, z1) $\wedge$
ARG1(z0, z2) $\wedge$
ARG2(z0, z4) $\wedge$
ARG0(z2, z4) $\wedge$
ARG1(z2, z3)

As one can see, AMR builds on top of the PropBank framesets, similar to SRL (refer to same example in Section 2.4). Differently from SRL though, AMR does not explicitly align the nodes of the graph with the word in the sentence. This "decoupling" from the syntactic structure of a sentence, allows more freedom in handling cases of syntax-semantic mismatches, and leads to encoding different syntactic realizations of the same meaning using the same structure. For instance, words that do not contribute to the meaning of a sentence are left out of the AMR annotation or are collapsed into single relations, e.g., discontinuous constructions such as "if ... then" can be collapsed into a single relation `:condition`. In addition, each AMR concept node is labeled with a variable name. Variable names are devoid of meaning, yet they are important especially for tracking *coreference*. Indeed, when a variable appears multiple times, all occurrences denote the same concept. For example, `patient` to which we assign the variable `p`, plays both the roles of *hearer* and *taker* for `tell-01` and `take-01` predicates, respectively. Thus, the node for `patient` has more than one incoming edge. This phenomenon is called reentrancy and it is an important aspect of meaning not covered by SRL. Moreover, AMR is a hierarchical structure, with the root node `(t / tell-01)` denoting the focus of the graph, which binds the contents of an AMR into a single, traversable directed graph.

This example shows only the main predicate-argument structure as captured by an AMR.

However, AMR includes several phenomena which make *parsing* challenging and that are usually handled through rules and intrinsic heuristics in the literature. We direct the reader to AMR guidelines[5] for a detailed overview of AMR components and specifications.

2.5.1 Multilingual AMR

Even though AMR has been initially designed to represent the meaning of sentences in the English language, and was stated not to be an interlingua [Banarescu et al., 2013], it gained quickly the attention of researchers, and several works attempted to adjust it for applicability across languages. The development of PropBank in other languages (see Section 2.3), allows defining specifications of AMR in the languages it is available. The largest non-English AMR corpus available is the Chinese AMR [Li et al., 2016, CAMR]. The authors developed specifications and annotated the *Little Prince* novel with Chinese AMR graphs.[6]

In fact, multilingual AMR has mainly been studied within the scope of annotation analysis in Czech and Chinese [Xue et al., 2014; Hajič et al., 2014], in Portuguese [Sobrevilla Cabezudo and Pardo, 2019], and in Spanish [Migueles-Abraira et al., 2018]. However, these works point out the limitations of AMR as an interlingua, and consider them partly due to the distinctions in the underlying resources and structural divergences among languages. More recently, Zhu et al. [2019a] and Van Gysel et al. [2021] worked at the formalism level; the former suggest simplifying AMR so as to express only predicate roles and linguistic relations in a sentence, in order to be able to apply it across languages. On the contrary, the latter design UMR as an extension of AMR, which i) adds aspect and scope, ii) includes temporal and modal dependencies at sentence- and document-level, iii) adapts AMR to a cross-lingual formalism allowing language-specific distinctions with extra relations.

2.5.2 Cross-lingual AMR

A peculiar feature of the AMR formalism is that it aims at abstracting away from word forms. AMR graphs are *unanchored*, i.e., the linkage between tokens in a sentence and nodes in the corresponding graph is not explicitly annotated. The fact that word order and morpho-syntactic variations account for much of the cross-linguistic variations, coupled

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

[6] https://www.cs.brandeis.edu/~clp/camr/camr.html

with the feature of being agnostic about how to derive meanings from strings, makes AMR particularly suitable for representing semantics cross-lingually. Damonte and Cohen [2018] proposed the usage of English-centric AMR graphs to represent sentences in any language, i.e., with the meaning representation associated with their English translation. For instance, the sentences below would be represented with the graph corresponding to the English sentence, shown in Figure 2.2:

English: *The doctor told the patient to take the medicine.*
Italian: *Il dottore ha detto al paziente di prendere la medicina.*
Spanish: *El doctor le dijo al paciente que se tomara la medicación.*
Albanian: *Doktori i tha pacientit që të marri mjekimin.*

2.5.3 Approaches to AMR

In this Section, we briefly overview the start-of-the-art in AMR parsing, AMR generation, and cross-lingual AMR parsing.

English AMR parsing. State-of-the-art results in AMR parsing have been previously attained by approaches that use more complex and multi-modular architectures. These combine seq-to-seq methods with graph-based algorithms in either two-stage [Zhang et al., 2019a] or incremental one-stage [Zhang et al., 2019b; Cai and Lam, 2020a] procedures. Moreover, they integrate similar processing pipelines and additional features including fine-grained graph recategorization [Zhang et al., 2019a,b; Zhou et al., 2020b; Cai and Lam, 2020a], which all contribute significantly to the performances achieved. On the other hand, simple seq-to-seq approaches model AMR parsing as a transduction of the sentence into a linearization of the AMR graph. Due to their end-to-end nature, such approaches are appealing for this task. However, since seq-to-seq-based approaches are data-hungry, their performances for AMR parsing have, until recently, been rather unsatisfactory, due to the relatively small amount of annotated sentence-AMR pairs. To overcome data sparsity, various different techniques have been employed by early seq-to-seq approaches: self-training using unlabeled English text [Konstas et al., 2017], character-level networks [van Noord and Bos, 2017], and concept recategorization as a preprocessing step to reduce the open vocabulary components, e.g., named entities and dates [Peng et al., 2017; van Noord

and Bos, 2017; Konstas et al., 2017]. Moreover, seq-to-seq-based models often incorporate features such as lemma, POS, or NER tags, as well as syntactic and semantic structures [Ge et al., 2019].
In our work [Bevilacqua et al., 2021a; Blloshmi et al., 2021a], we wear off the complexities of the English AMR parsing relying almost exclusively on seq-to-seq, disposing of the need for extra features, and employing a lightweight postprocessing pipeline, only for ensuring graph validity. Nonetheless, we significantly outperform previous state-of-the-art approaches that, we recall, feature complexities in architecture and pre- and postprocessing pipelines. Additionally, we show that the extensive recategorization techniques, while boosting performance on the traditional in-domain benchmarks, are harmful in the Out-of-Distribution (OOD) setting. Moreover, while other approaches have employed pretrained *encoders*, such as BERT [Devlin et al., 2019], in order to have powerful features for a parsing architecture [Zhang et al., 2019a,b; Cai and Lam, 2020a], we are the first to show that pretrained *decoders*, too, are beneficial for AMR parsing, even though the pretraining only involves English, and does not include formal representations (Chapter 4).

English AMR generation. AMR generation has been performed with two main approaches: explicitly encoding the graph structure in a graph-to-text transduction fashion through graph neural network models [Song et al., 2018; Beck et al., 2018; Damonte and Cohen, 2019; Zhu et al., 2019b; Cai and Lam, 2020b; Yao et al., 2020], or as a purely seq-to-seq task through AMR graph linearization [Konstas et al., 2017; Mager et al., 2020]. Recent graph-based approaches rely on Transformers to encode AMR graphs [Zhu et al., 2019b; Cai and Lam, 2020b; Wang et al., 2020; Song et al., 2020; Yao et al., 2020]. The model of Mager et al. [2020] is a pretrained Transformer-based decoder-only model fine-tuned on a sequential representation of the AMR graph.
In our work [Bevilacqua et al., 2021a; Blloshmi et al., 2021a], we use an encoder-decoder architecture, which is more suitable for handling conditional generation and casts AMR generation as symmetric to AMR parsing, therefore disposing of the need for a task-specific model (Chapter 4).

Cross-lingual AMR parsing. Cross-lingual AMR *parsing*, instead, has received relatively less attention. This is largely attributable to the lack of training data and evaluation benchmarks in languages other than English. Damonte and Cohen [2018] propose the first cross-lingual AMR parser and, more recently, they released their proposed cross-lingual AMR evaluation benchmark [Damonte and Cohen, 2020]. The authors adapt a transition-based English AMR parser [Damonte et al., 2017] for cross-lingual AMR parsing, which is trained on silver annotated data. However, the performances it has achieved are not satisfying in terms of Smatch score [Cai and Knight, 2013], mostly as a result of concept identification errors, which in turn are directly related to the usage of noisy word-to-node alignments projected from English. Throughout the literature English AMR parsers commonly rely on AMR alignments which are automatically created using heuristics [Flanigan et al., 2014], or on pretrained aligners [Pourdamghani et al., 2014; Liu et al., 2018], treated as latent variables of the model [Lyu and Titov, 2018], or implicitly modeled through *source-copy* mechanisms [Zhang et al., 2019a]. These alignments, however, take advantage of the fact that AMR nodes and English words are highly related.[7] This dependency is therefore not suitable for cross-lingual parsing since similarity between words in the sentences and concepts in the graph does not hold at large.

In our work [Blloshmi et al., 2020], we propose a cross-lingual parser that disposes of explicit and implicit AMR alignments using a seq-to-seq model for concept identification and achieves significantly higher performance on all the tested languages (Chapter 5).

After its publication, XL-AMR has been followed by a relatively large number of works with significant improvements in cross-lingual AMR parsing. Briefly, these advancements include joint training of machine translation and semantic parsing tasks for zero-shot cross-lingual AMR parsing [Procopio et al., 2021], leveraging robust contextualized word embeddings to improve the foreign-text-to-English-AMR alignments [Sheth et al., 2021], using bilingual input (paired with English) [Cai et al., 2021b], learning a multilingual AMR parser by using an existing English parser as its teacher [Cai et al., 2021a], or leveraging translation models to first translate into English and then parse into AMR [Uhrig et al., 2021].

[7] In AMR 2.0 roughly 60% of the nodes are English words. In addition, PropBank predicates are often similar to English words, e.g., one can heuristically align `publish-01` to `publish`.

2.6 Semantically-enhanced Applications

Although established understanding has it that semantic structures ought to improve text understanding for NLP tasks such as question answering and machine translation, the early work done towards deeming semantics beneficial for downstream applications has been inconclusive or complementary. There were at least two good reasons for this outcome; First, until a few years ago, the performance of semantic parsers has been unsatisfactory, especially when applicable to out-of-domain data. Second, different applications require common-sense information, which is not necessarily comprised within the parsed semantic structures.

More recently, instead, due to the advancements in both SRL and AMR research, numerous efforts have found them to be beneficial in a wide range of downstream applications. Indeed, SRL has been proven beneficial not only in Natural Language Processing but also in Computer Vision, including: Question Answering [Shen and Lapata, 2007], Visual Semantic Role Labeling [Gupta and Malik, 2015], Situation Recognition [Yatskar et al., 2016], and Machine Translation [Marcheggiani et al., 2018b]. The latter incorporates information about the predicate-argument structure of a sentence into a neural machine translation using Graph Convolutional Neural networks, similar to previous work, which instead includes syntactic information [Bastings et al., 2017]. Indeed, their results show that semantics is more beneficial than syntax for neural machine translation.

Furthermore, since AMR includes within its formalism the predicate-argument structure captured by SRL, it features an even more comprehensive range of applications. Indeed, AMR's flexibility has resulted in promising improvements in Machine Translation [Song et al., 2019c], Text Summarization [Hardy and Vlachos, 2018; Liao et al., 2018], Paraphrase Detection [Issa et al., 2018], Entity Linking [Pan et al., 2015], Human-Robot Interaction [Bonial et al., 2020a], Information Extraction [Rao et al., 2017], and more recently, Question Answering [Bonial et al., 2020b; Lim et al., 2020; Kapanipathi et al., 2021]. Lim et al. [2020] present an interesting combination of AMR and ConceptNet [Speer et al., 2017a], an external commonsense knowledge graph, to form the so-called AMR-ConceptNet-Pruned (ACP) graph. This new semantically-rich graph is then exploited to interpret and predict the correct answer for the CommonsenseQA task [Talmor et al., 2019], achieving higher performance than baselines that do not use symbolic meaning representations. Similarly, Kapanipathi et al.

[2021] convert an AMR to logical knowledge graph triples and enrich it with explicit links to entities in the knowledge graph. Then, they perform knowledge-based question answering and achieve state-of-the-art performances in multiple benchmarks. The authors argue that the usage of symbolic meaning representations is beneficial as the task of understanding natural language questions is delegated to AMR parsers. In addition, the abilities of AMR to deal with syntactic idiosyncrasies and to handle complex sentence structures, such as multi-hop questions or imperative statements, make the question-answering system more robust to changes in the input questions. Nevertheless, Kapanipathi et al. [2021] discuss a set of challenges coming with the integration of AMR parsers in downstream applications, mainly related to the performance of the existing parser across different domains.

While these findings make us optimistic about getting closer to machine understanding through conceptual representations, they stress the importance of designing higher-performing Semantic Parsing approaches across domains and languages. With this in mind, we propose a more challenging evaluation setting for AMR parsing and generation for a more realistic assessment of the generalizability of future approaches (Chapter 4).

∼ *This page was intentionally left blank* ∼

Chapter 3

End-to-End SRL as Sequence Generation

Abstract

Despite the recent great success of the seq-to-seq paradigm in Natural Language Processing, the majority of current studies in Semantic Role Labeling (SRL) still frame the problem as a sequence labeling task. In this Chapter we go against the flow and propose GSRL (*Generating Senses and RoLes*), the first seq-to-seq model for end-to-end SRL. Our approach benefits from recently-proposed decoder-side pretraining techniques to generate both sense and role labels for all the predicates in an input sentence at once, in an end-to-end fashion. Evaluated on standard gold benchmarks, GSRL achieves state-of-the-art results in both dependency- and span-based English SRL, proving empirically that our simple generation-based model can learn to produce complex predicate-argument structures. Finally, we propose a framework for evaluating the robustness of an SRL model in a variety of synthetic low-resource scenarios which can aid human annotators in the creation of better, more diverse, and more challenging gold datasets. We release GSRL at `https://github.com/SapienzaNLP/gsrl`.

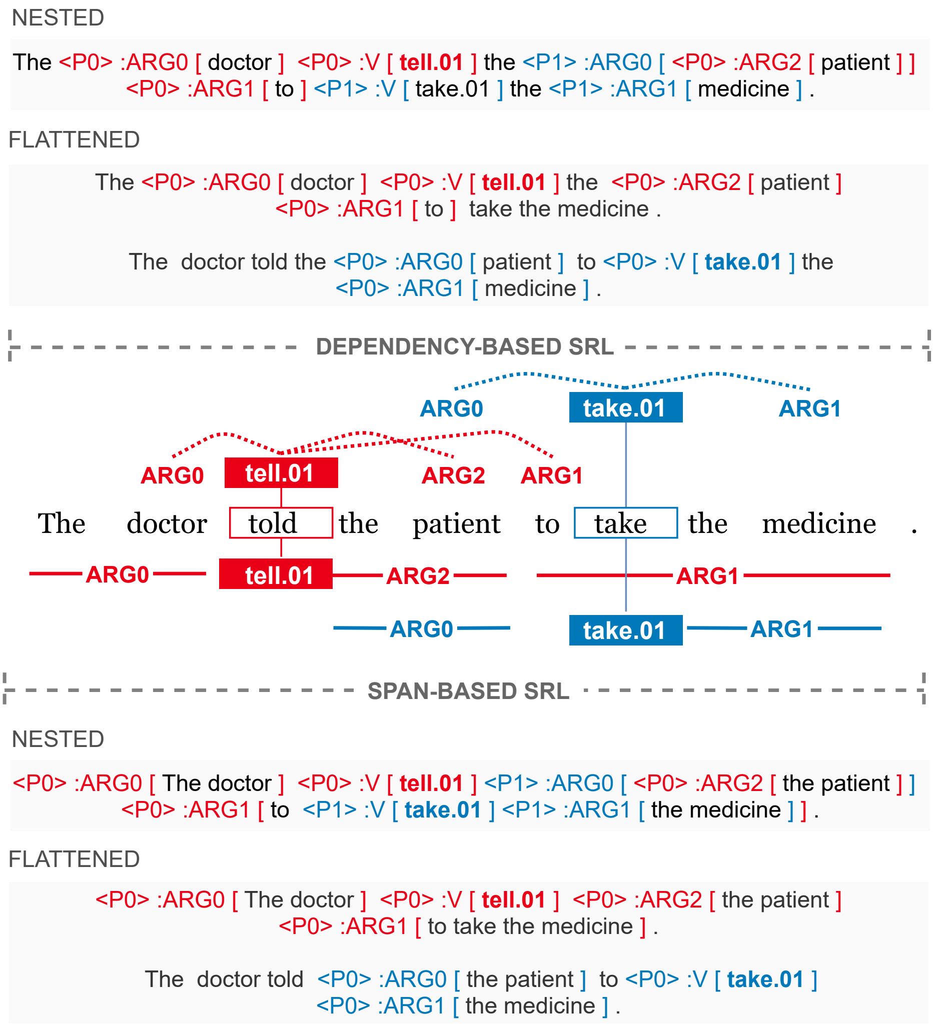
Source: This Chapter is based on our IJCAI 2021 paper [Blloshmi et al., 2021b]: Generating Senses and RoLes: An End-to-End Model for Dependency- and Span-based Semantic Roles Labeling.

3.1 Overview

Semantic Role Labeling (SRL) approaches revolved predominantly around sequence labeling paradigm, with only a small handful of attempts at tackling the task in a seq-to-seq fashion. While sequence labeling approaches represented more complex and task-specific architectures with respect to general-purpose seq-to-seq architectures, the existing seq-to-seq approaches to SRL did not handle the predicate disambiguation step, thus not being end-to-end, and also used to take advantage of encoder pretraining only, leaving the decoder-side pretraining unexplored. In addition, even though sequence labeling models outperformed the existing seq-to-seq approaches, it was unclear why most of the recent models started converging in the same performance pool. Indeed, casting SRL as a sequence generation problem comes with the advantage of being a general-purpose architecture, which in turn, makes minimal assumptions on the structure of the data. This formulation might allow future extensions to more complex semantic structures generation via multitask learning strategies, e.g., combining SRL with other related task learning such as AMR parsing. Furthermore, recent pretrained encoder-decoders have shown advancements in different NLP tasks across different domains. To this end, modeling SRL as a seq-to-seq learning problem allows for better exploitation of the knowledge encoded in the weights of powerful pretrained models. In this context, to address the gaps in SRL research, we presented GSRL [Blloshmi et al., 2021b].

In this Chapter, we detail GSRL (*Generating Senses and RoLes*), a novel end-to-end approach to generating both predicate senses and semantic roles [Blloshmi et al., 2021b]. The contributions of this work are:

- We introduce the first seq-to-seq model for end-to-end SRL, tackling predicate sense disambiguation, argument identification and argument classification as a single generation task;
- We demonstrate that seq-to-seq learning can achieve state-of-the-art results, previously attained only by sequence labeling approaches, in multiple gold benchmarks for both dependency- and span-based English SRL;
- We compare different strategies to represent predicate-argument relations and generate structured, graph-like sense and role annotations, analyzing their characteristics;
- Motivated by the convergence in the performance of recent SRL systems, we propose a framework to i) evaluate future innovations in more challenging settings and ii) aid the creation of new SRL datasets.



NESTED

The <P0> :ARG0 [doctor] <P0> :V [**tell.01**] the <P1> :ARG0 [<P0> :ARG2 [patient]] <P0> :ARG1 [to] <P1> :V [take.01] the <P1> :ARG1 [medicine] .

FLATTENED

The <P0> :ARG0 [doctor] <P0> :V [**tell.01**] the <P0> :ARG2 [patient] <P0> :ARG1 [to] take the medicine .

The doctor told the <P0> :ARG0 [patient] to <P0> :V [**take.01**] the <P0> :ARG1 [medicine] .

NESTED

<P0> :ARG0 [The doctor] <P0> :V [**tell.01**] <P1> :ARG0 [<P0> :ARG2 [the patient]] <P0> :ARG1 [to <P1> :V [**take.01**] <P1> :ARG1 [the medicine]] .

FLATTENED

<P0> :ARG0 [The doctor] <P0> :V [**tell.01**] <P0> :ARG2 [the patient] <P0> :ARG1 [to take the medicine] .

The doctor told <P0> :ARG0 [the patient] to <P0> :V [**take.01**] <P0> :ARG1 [the medicine] .

Figure 3.1. Example of a sentence with two predicates: dependency-based SRL (upper part) and span-based SRL (lower part) and their corresponding *nested* and *flattened* linearizations.

3.2 Methodology

3.2.1 SRL as a Sequence-to-Sequence Task

We revisit the *seq-to-seq* formulation by Daza and Frank [2018] for PropBank-based SRL and put forward a generalized formulation that is able to handle not only semantic role labels but also predicate sense labels. Formally, given a sentence $\mathbf{s} = \langle w_1, w_2, \ldots, w_{|\mathbf{s}|} \rangle$ where each word w_i belongs to either the vocabulary of words V^{W} or a vocabulary of special tokens V^{ST}, the model is required to generate a sequence $\mathbf{o} = \langle o_1, o_2, \ldots, o_{|\mathbf{o}|} \rangle$ where each token o_i belongs to either the input sentence s, the vocabulary of special tokens V^{ST}, the semantic role vocabulary V^{SR}, or the predicate sense vocabulary V^{Ps}.

As shown in Figure 3.1, we propose two strategies for generating the predicate-argument relations:

- *Flattened* linearization in which the model is required to generate a separate sequence $\mathbf{o}_p$ for each predicate p in $\mathbf{s}$, where $\mathbf{o}_p$ contains the sense and role labels only for p;
- *Nested* linearization in which the model is required to generate a single sequence $\mathbf{o}$ containing the sense and role labels for all the predicates in $\mathbf{s}$.

If we exclude predicate sense labels from the generated sequence $\mathbf{o}$, our *flattened* linearization strategy is similar to that of Daza and Frank [2018] and can be considered as a simplified or "unrolled" semantic structure of our *nested* linearization. We build the *nested* linearization in left-to-right order, i.e., the role label related to the first occurring predicate is positioned innermost, and the subsequent roles encapsulate all the previously seen labels of an argument. We argue that the semantics of the *nested* linearization, while being more complex to learn, comes with the advantage of providing the entire predicate-argument structure of the input sentence $\mathbf{s}$ at once, reducing the overhead of generating a number of output sequences equal to the number of predicates in $\mathbf{s}$, and thus being more practical for an end system.

3.2.2 GSRL Model

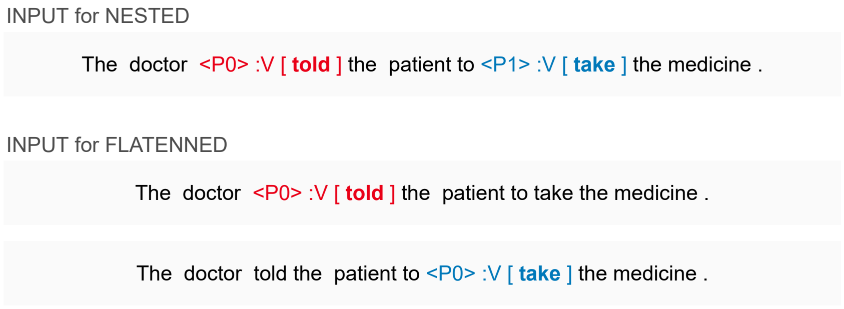
Given the above definition of seq-to-seq SRL, we formally frame the task as a conditional generation problem in which we want to maximize the probability $P(\mathbf{g}|\mathbf{t})$ of generating the tokenization $\mathbf{g} = \langle g_1, g_2, \ldots, g_i, \ldots, g_{|\mathbf{g}|}\rangle$ of the output linearization $\mathbf{o}$ conditioned on the tokenization $\mathbf{t} = \langle t_1, t_2, \ldots, t_{|\mathbf{t}|}\rangle$ of the input sentence $\mathbf{s}$:

$$P(\mathbf{g}|\mathbf{t}) = \prod_{i=2}^{|\mathbf{g}|} P(g_i \mid \mathbf{g}_{1:i-1}, \mathbf{t}) \tag{3.1}$$

where g_1 is the artificially added start token <s>, g_i is the i-th element (token, special token, sense, or role) of the generated output sequence $\mathbf{g}$ and $\mathbf{g}_{1:i-1} = \langle g_1, g_2, \ldots, g_{i-1}\rangle$. Therefore, the probability $P(\mathbf{g}|\mathbf{t})$ of the linearized predicate-argument structure $\mathbf{g}$ for the given sentence $\mathbf{t}$ is computed as the product of the probability of generating each token g_i of $\mathbf{g}$ in an autoregressive fashion.

The GSRL model architecture builds on top of BART [Lewis et al., 2020], a recently proposed denoising autoencoder for seq-to-seq learning. BART can be seen as a generalization of several modern language models from BERT (due to the bidirectional encoder) to GPT (with the left-to-right decoder), and it was found to be particularly effective in a wide range of Natural Language Understanding tasks, including tasks that involve complex structured outputs such as semantic parsing [Bevilacqua et al., 2021a]. Following BART, our model architecture is based on a Transformer-based neural machine translation architecture [Vaswani et al., 2017a], with 12 stacked Transformer layers for both the encoder and the decoder. However, rather than training GSRL to learn to maximize the conditional probability shown in Equation 3.1 from scratch, we warm-start the model with the weights of BART, which brings two significant advantages. First, GSRL inherits the capability of BART to denoise artificially-corrupted sentences and generate an output sequence that, while (partially) overlapping with the input sequence, can have a different length. This is beneficial to our setting, since the input sequence fed into the model can be seen as a corrupted sentence where the sense and role annotations have been removed. Second, GSRL can take advantage of the world of knowledge coming from the massive amounts of text BART has been pretrained on. Indeed, the original training corpus for BART is composed of five English-language corpora of varying sizes and domains, containing books, stories, news, web content and Wikipedia articles, and thus providing a wealth of information that could otherwise be missing from standard SRL datasets, given their relatively small size.

Vocabulary. We start from the vocabulary of BART which, thanks to its BPE tokenization, includes V^{W}, and extend it by adding i) the set V^{Ps} of PropBank predicate sense labels, e.g., `tell.01` and `take.01`, ii) the set V^{SR} of PropBank semantic role labels, e.g., `:ARG0` and `:ARGM-NEG`, and iii) the set V^{ST} of special tokens to distinguish between verbal and nominal predicates, i.e., `:V` and `:N` respectively, and to identify the predicates in the sentence, i.e., <Pi>, where i is the order of the predicate in the input sentence from left to right. At the input level, we make sure that the BPE tokenizer does not split the additional tokens. Therefore, adding these task-specific atomic tokens to the vocabulary allows for a more compact linearized SRL structure. Finally, we randomly initialize the embeddings of the additional tokens and update their values during training.



INPUT for NESTED

The doctor <P0> :V [**told**] the patient to <P1> :V [**take**] the medicine .

INPUT for FLATENNED

The doctor <P0> :V [**told**] the patient to take the medicine .

The doctor told the patient to <P0> :V [**take**] the medicine .

Figure 3.2. Example of a sentence with two predicates: input sequence for *nested* target sequence (upper part) and *flattened* target sequence (lower part).

3.2.3 Pre- and Postprocessing

Preprocessing. The input sentence is preprocessed differently depending on the linearization strategy – *flattened* or *nested* – chosen to train the GSRL model. Before feeding an input sentence into the model, we indicate each predicate with a special token <Pi> which guides the model towards learning to distinguish between different predicates and to specifically generate the argument roles for each of them, where $i = 0$ in the *flattened* linearization and $0 \leq i < n_p$ in the *nested* linearization, with n_p being the total number of predicates in the sentence. A visualisation of the inputs is shown in Figure 3.2. In the flattened linearization setting, the input sentence is repeated n_p times, i.e., it would be preprocessed twice. When the GSRL model is trained to generate *nested* linearizations, the input sentence is preprocessed to indicate all the predicates at once.

Postprocessing. As opposed to sequence labeling approaches, our seq-to-seq model is not only trained to produce sense and role labels, but also to autoregressively regenerate the words of the input sentence. Therefore, an output sequence is valid only if the following two conditions are met: i) its words can be aligned to the words of the input sequence, and ii) all the predicate-argument structures follow the PropBank annotation guidelines. Indeed, in order to enforce valid predicate-argument structures during the annotation process, PropBank-based SRL requires human annotators to follow a set of guidelines which state that core roles (`ARG0`, `ARG1`, etc.) must appear at most once for each predicate, two arguments of the same predicate must not overlap, reference roles (`R-ARG0`, `R-ARGM-TMP`, etc.) can only appear if they refer to an existing core role in the sentence, and continuation roles

(`C-ARG0`, `C-ARGM-TMP`, etc.) can only appear after the core role they refer to, *inter alia*. For the sake of simplicity, our model is not explicitly constrained to generate a valid predicate-argument structure, and we only adopt the following simple heuristics to postprocess an output sequence:

- In span-based SRL, we close at most one unenclosed argument span, positioning the closing bracket so that there are no two overlapping arguments for the same predicate;
- In span-based SRL, if more than one span is unenclosed, we discard all the spans;
- In both dependency- and span-based SRL, if two arguments of the same predicate overlap, we discard all the arguments for the sentence.

Previous studies have shown that explicitly enforcing PropBank constraints leads to more accurate predictions [Li et al., 2019], but in this work we focus on unconstrained generation and leave constrained generation for future work. For instance, methods used in previous works that enforce validity constraints for other SRL modeling paradigms [Das et al., 2014; Täckström et al., 2015; Li et al., 2020], could be interesting to integrate within our seq-to-seq model.

3.3 Experiments

3.3.1 Evaluation Benchmarks

We train and evaluate GSRL on the standard splits of the English datasets provided as part of the CoNLL-2009 [Hajič et al., 2009] and CoNLL-2012 [Pradhan et al., 2012] shared tasks, which rapidly became two standard benchmarks for dependency- and span-based SRL, respectively. While CoNLL-2009 is mainly composed of finance-related documents coming from the Wall Street Journal, CoNLL-2012 is a varied collection of news, conversations and magazine articles. Additionally, CoNLL-2009 includes an out-of-domain test set containing excerpts from the Brown Corpus.

Data statistics. We define the semantic complexity of a dataset as the number of predicate-argument relations that appear in each sentence on average. In CoNLL-2012, we observe that around 70% of the sentences are annotated with at most 5 role labels and 3 predicates, with an average of 2.8 predicates per sentence. However, this is not the case in CoNLL-2009 where only 20% of the sentences contain at most 5 role labels, and only 40% feature at most 3 predicates. In fact, CoNLL-2009 has an average of 4.7 predicates per sentence, almost twice the number compared to CoNLL-2012. These statistics suggest that the semantic complexity of CoNLL-2009 is higher than that of CoNLL-2012, and thus it is to be expected that the predicate-argument structures in CoNLL-2009 should be more complex, making the *nested* linearizations deeper and more difficult to learn.

3.3.2 Evaluation Metrics

In the following Sections, we report the scores of the official scorers provided as part of the CoNLL shared tasks to measure the performance of a participating system. More specifically, the standard evaluation script for span-based PropBank-style SRL is the CoNLL-2005 scorer[1] which computes precision, recall and F1 score of the semantic roles. For dependency-based PropBank-style SRL we use the CoNLL-2009 scorer[2] which takes into account both sense and role labels to compute what is referred to as "semantic" precision and recall:

$$P_{\text{SEM}} = \frac{\text{TP}^{\text{pred}} + \text{TP}^{\text{role}}}{\text{N}^{\text{pred}} + \text{TP}^{\text{role}} + \text{FP}^{\text{role}}}$$

$$R_{\text{SEM}} = \frac{\text{TP}^{\text{pred}} + \text{TP}^{\text{role}}}{\text{N}^{\text{pred}} + \text{TP}^{\text{role}} + \text{FN}^{\text{role}}}$$

where TP, FP and FN are the true positives, false positives and false negatives, respectively, while N^{pred} is the total number of predicates.

3.3.3 Training and Tuning

We train two main model configurations using the *flattened* and *nested* linearizations, $\text{GSRL}_{\textit{flattened}}$ and $\text{GSRL}_{\textit{nested}}$ hereafter. For both variants, their weights are warm-started using $\text{BART}_{\text{large}}$ (406M parameters) from the Transformers library.[3] Differently from vanilla BART, we increase the dropout rate between the Transformer layers from 0.1 to 0.25 and

[1] cs.upc.edu/∼srlconll/soft.html

[2] ufal.mff.cuni.cz/conll2009-st/scorer.html

[3] huggingface.co/transformers/model_doc/bart.html

PARAMETER	PICK	SEARCH SPACE
Training		
Learning Rate (LR)	$5 * 10^{-5}$	1/5/10/50 $*10^{-5}$
LR Scheduling	constant	-
Loss	Cross-entr.	-
Betas	0.9, 0.999	-
Epochs	20	[10, 20]
Dropout	0.25	0.1 to 0.25, (0.05)
Weight Decay	0.004	0.001 to 0.01, (+0.001)
Gradient Accumulation	10	[1, 5, 10, 15, 20]
Prediction		
Beam size	1	[1, 5]

Table 3.1. GSRL hyperparamter values and search space.

we do not penalize the model for the generation of repeated *ngrams*, e.g., multiple closing brackets. In Table 3.1 we report the hyperparameters space of GSRL. We pick the parameters using random search with 5 trials in the search space indicated in the third column. Finally, we select the best model based on its F1 score on the development dataset. At prediction time we perform only greedy decoding, since beam searching did not show improvements in our preliminary experiments. Each GSRL model is trained for 20 epochs with a batch size of 800 tokens, using the RAdam [Liu et al., 2020a] optimizer with a fixed learning rate of 1×10^{-5} and gradient accumulation every 10 batches. The training process is carried out on a single GPU (Nvidia GeForce GTX 1080Ti): GSRL$_{\textit{flattened}}$ requires 30 and 40 hours of training time on CoNLL-2009 and CoNLL-2012, respectively, while GSRL$_{\textit{nested}}$ requires 11 and 20 hours on CoNLL-2009 and CoNLL-2012, respectively.

3.3.4 Comparison Systems

The vast majority of the recent advances in SRL come from sequence labeling approaches, which currently represent the state of the art in both span- and dependency-based SRL. Therefore, we mainly compare our seq-to-seq model against the recent innovations proposed by such sequence labeling models, but also to the few existing seq-to-seq approaches to the task. As such, the comparison systems are divided into these two paradigms. First, we include sequence labeling approaches that:

i) jointly learn SRL and syntax [Cai and Lapata, 2019b];

ii) iteratively refine the output SRL labels [Lyu et al., 2019];

iii) devise a set of syntactic “supertags” [Kasai et al., 2019];

iv) learn predicate-argument interactions through capsule networks [Chen et al., 2019];

v) better exploit the knowledge of language models [Shi and Lin, 2019; Conia and Navigli, 2020];

vi) model syntactic dependencies with graph convolutions [Marcheggiani and Titov, 2020].

As per seq-to-seq approaches, we compare with Daza and Frank [2018, 2019], who proposed, to the best of our knowledge, the currently best-performing seq-to-seq models for SRL. However, GSRL significantly differs from their architectures which i) are not able to handle multiple predicates at once, and ii) do not address predicate sense disambiguation, i.e., they are not end-to-end.

For completeness, in our experiments we include two other challenging variations of GSRL. In particular, we challenge GSRL$_{nested}$ to perform the *predicate identification* step (GSRL$^{\text{PI}}_{flattened}$, hereafter). For this evaluation, the input sentence is not preprocessed, i.e., the input sequence does not contain any predicate tags (see Section 3.2.3). In addition, we observe the behavior of GSRL$_{flattened}$ when no BART pretraining is used (GSRL$^{\text{S}}_{flattened}$, hereafter), i.e., the architecture is trained from scratch on respective CoNLL datasets for dependency- and span-based SRL.

3.4 Results

3.4.1 Dependency-based SRL Results

Table 3.2 summarizes the results on dependency-based SRL in the English in-domain test of CoNLL-2009. Even though GSRL is also tasked to generate predicate sense labels, GSRL$_{flattened}$ significantly surpasses the previously best-performing seq-to-seq model of Daza and Frank [2019] by 1.6% in F$_1$ score (17% decrease in error rate).[4] While both

[4] Daza and Frank [2019] rely on a separate system trained on a larger amount of sentences in order to output predicate sense labels.

CoNLL-2009 – In domain	P	R	F_1
Sequence labeling models			
Cai and Lapata [2019b]	90.9	89.1	90.0
Lyu et al. [2019]	–	–	90.1
Kasai et al. [2019]	90.3	90.0	90.2
Li et al. [2019]	89.6	91.2	90.4
He et al. [2019]	90.4	91.3	90.9
Chen et al. [2019]	90.7	91.4	91.1
Cai and Lapata [2019a]	91.7	90.8	91.2
Shi and Lin [2019]	92.4	92.3	92.4
Conia and Navigli [2020]$_{\text{XLM-R}}$	92.2	92.6	92.4
Conia and Navigli [2020]$_{\text{BERT}}$	92.5	92.7	92.6
Sequence-to-sequence models			
Daza and Frank [2019]	–	–	90.8
GSRL$_{nested}$	91.8	86.5	89.0
GSRL$_{flattened}$	92.9	92.0	92.4

Table 3.2. Results on the English in-domain test set of the CoNLL-2009 task for dependency-based SRL. P: precision. R: recall.

systems take advantage of pretrained encoders (BART and ELMo), GSRL also exploits the pretrained decoder of BART, which allows for superior performance. Moreover, when compared to state-of-the-art *sequence labeling* approaches [Shi and Lin, 2019; Conia and Navigli, 2020], GSRL$_{flattened}$ shows competitive results, with an F$_1$ score that is either matching or not statistically different. It is interesting to note that, while GSRL$_{nested}$ is tasked to learn semantic structures that can be an order of magnitude more complex than those learnt by its GSRL$_{flattened}$ counterpart, the resulting difference in performance is not as large as one may expect, and the training process is more than 60% faster. However, the considerably lower recall shown by GSRL$_{nested}$ empirically confirms the complexity of identifying and generating longer sequences of predicate and role labels, especially when a single word is enclosed by multiple labels, i.e., it is an argument for multiple predicates. For example, in <P2> `:ARG0 [`<P1> `:ARG0 [`<P0> `:ARG0 [`<P0> `:N [chairman.01]]]]`, the predicate `chairman.01` plays 4 roles, therefore GSRL$_{nested}$ fails to generate all of them.

In Table 3.3 we compare GSRL variations. First, GSRL$^{\text{PI}}_{nested}$ attains 5.8 F$_1$ points less when compared to GSRL$_{nested}$. Indeed, the largest drop in performance is due to the low recall, which indicates that the system is not able to either identify all the predicates, or

CoNLL-2009 – In domain	P	R	F_1
GSRL$_{nested}$	91.8	86.5	89.0
GSRL$_{flattened}$	92.9	92.0	92.4
GSRL$^{\mathrm{PI}}_{nested}$	86.9	79.8	83.2
GSRL$^{\mathrm{S}}_{flattened}$	86.7	84.4	85.5

Table 3.3. Results of GSRL variations on the English in-domain test set of the CoNLL-2009 task for dependency-based SRL. P: precision. R: recall.

CoNLL-2009 – Out of domain	P	R	F_1
Sequence labeling models			
Li et al. [2019]	–	–	81.5
Lyu et al. [2019]	–	–	82.2
Chen et al. [2019]	–	–	82.7
Conia and Navigli [2020]$_{\text{XLM-R}}$	–	–	85.2
Conia and Navigli [2020]$_{\text{BERT}}$	–	–	85.9
Sequence-to-sequence models			
Daza and Frank [2019]	–	–	84.1
GSRL$_{nested}$	85.0	80.1	82.5
GSRL$_{flattened}$	85.8	84.5	85.2

Table 3.4. Results on the English out-of-domain test of the CoNLL-2009 task for dependency-based SRL. P: precision. R: recall.

to appropriately handle the roles associated to each of them due to the missing predicate identifiers in the input sequence. Notice that this experiment cannot be performed with GSRL$_{flattened}$, for which we are constrained to use the identifier for the predicate of interest. Second, GSRL$^{\mathrm{S}}_{flattened}$, which is disadvantaged in that it is trained from scratch, achieves 6.9 F$_1$ points less that GSRL$_{flattened}$. However, this result is expected and confirms once more the benefits of pretrained power which allows the models to generalize better overall.

Furthermore, Table 3.4 reports the results in the English out-of-domain test of CoNLL-2009 where we observe a similar trend to the in-domain evaluation, with GSRL$_{flattened}$ significantly surpassing the previous *seq-to-seq* approach [Daza and Frank, 2019] and performing on a par with the state of the art [Conia and Navigli, 2020].

CoNLL-2012	P	R	F_1
Sequence labeling models			
Ouchi et al. [2018a]	87.1	85.3	86.2
Li et al. [2019]	85.7	86.3	86.0
Shi and Lin [2019]	85.9	87.0	86.5
Marcheggiani and Titov [2020]	86.5	87.1	86.8
Conia and Navigli [2020]	86.9	87.7	87.3
Sequence-to-sequence models			
Daza and Frank [2018]	–	–	75.4
GSRL$_{\textit{nested}}$	87.1	86.6	86.8
GSRL$_{\textit{flattened}}$	87.8	86.8	87.3

Table 3.5. Results on the English in-domain test set of the CoNLL-2012 gold benchmark for span-based SRL. P: precision. R: recall.

CoNLL-2012	P	R	F_1
GSRL$_{\textit{nested}}$	87.1	86.6	86.8
GSRL$_{\textit{flattened}}$	87.8	86.8	87.3
GSRL$^{\text{PI}}_{\textit{nested}}$	74.3	69.5	71.8
GSRL$^{\text{S}}_{\textit{flattened}}$	78.4	74.9	76.6

Table 3.6. Results of GSRL variations on the English test set of the CoNLL-2012 task for span-based SRL. P: precision. R: recall.

3.4.2 Span-based SRL Results

Table 3.5 summarizes the results on span-based SRL in the English test of CoNLL-2012. Similarly to CoNLL-2009, GSRL$_{\textit{flattened}}$ achieves state-of-the-art results in an area where *sequence labeling* approaches are currently predominant. In this setting, however, GSRL$_{\textit{flattened}}$ and GSRL$_{\textit{nested}}$ attain comparable performance, and they both surpass the *seq-to-seq* model of Daza and Frank [2018] by a large margin (more than 11.4% in F$_1$ score). The close gap between the two GSRL models can be explained by the lower semantic complexity of the sentences in CoNLL-2012 (see Section 3.3.1, Data Statistics), which results in easier SRL structures to be generated. Regarding the GSRL variations instead, in Table 3.6 we observe a large drop in performance when the model is required to perform predicate identification, i.e., GSRL$^{\text{PI}}_{\textit{flattened}}$, or when it is deprived of pretraining, i.e., GSRL$^{\text{S}}_{\textit{flattened}}$. In particular, GSRL$^{\text{S}}_{\textit{flattened}}$ scores around 10 F$_1$ points lower than GSRL$_{\textit{flattened}}$. This happens probably due the data from various collections included in CoNLL-2012, in which case

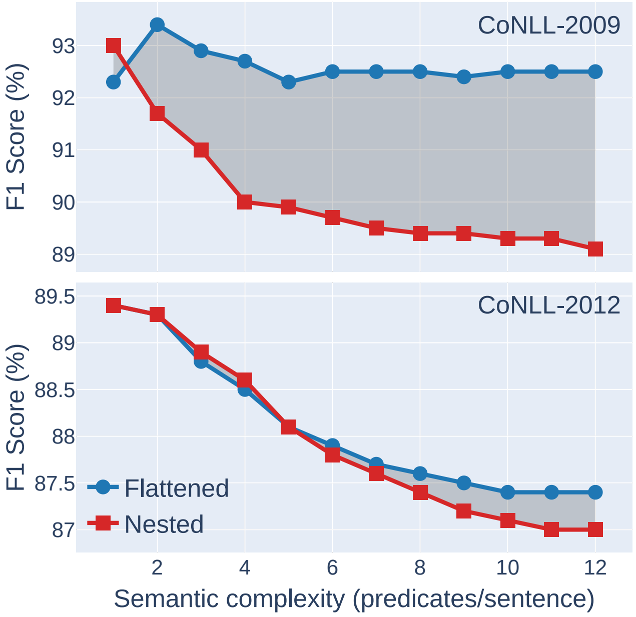


Figure 3.3. Results as the number of predicates per sentence becomes larger: the gap widens as the semantic complexity increases. This analysis shows that the semantic complexity of a sentence is the main culprit for the gap in performance between GSRL$_{flattened}$ and GSRL$_{nested}$.

it is more probable to observe unseen predicates at inference time than in CoNLL-2009, which instead is more domain specific. Owing to this, the model appears to benefit from the heterogeneous pretrained knowledge of BART more in CoNLL-2012 than CoNLL-2009. GSRL$^{\text{PI}}_{flattened}$ instead, performs poorly when compared with GSRL$_{flattened}$, achieving around 15 F$_1$ points less. This is even more unexpected when compared to its behaviors in the CoNLL-2009 test set, in which case the performance degrades by less than 6 F$_1$ points.

In summary, it is worth noting that in both span- and dependency-based SRL, while the F$_1$ score of GSRL is on a par with the best-performing *sequence labeling* approaches, GSRL always shows a higher precision.

3.5 Analysis

In what follows we propose an evaluation framework composed of a set of synthetic scenarios built from the CoNLL-2009 and CoNLL-2012 datasets. Our aim is two-fold: i) to better evaluate the behaviour of GSRL, or any other SRL system, and ii) to gain insights into what is needed for the creation of better training datasets or challenging benchmarks for SRL. In

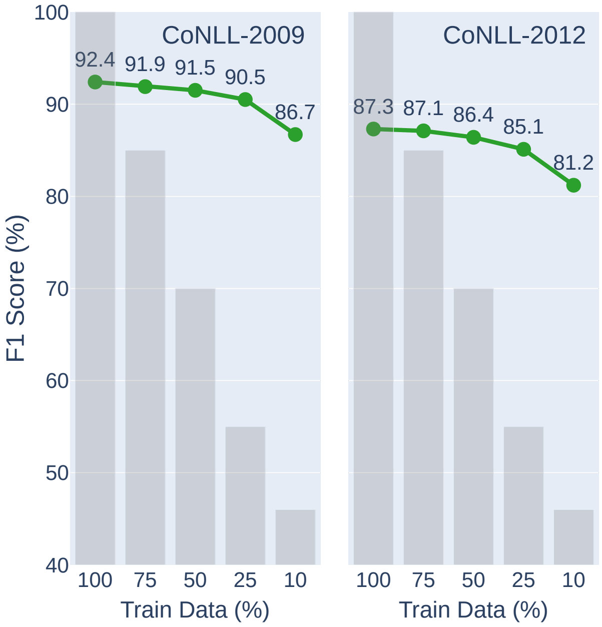


Figure 3.4. Results of GSRL*flattened* as the train data decreases: the margin between 100% and 25% is not large. This analysis shows that GSRL is robust to substantially smaller training datasets.

order to enable future comparisons with this work, we release our evaluation framework at `https://github.com/SapienzaNLP/gsrl`.

Test down-sampling: Semantic complexity. We observe the difference in performance between GSRL*flattened* and GSRL*nested* when including increasingly complex sentences in an initially empty test set. To this end, we build 12 test sets from both CoNLL-2009 and CoNLL-2012 by selecting each sentence according to its semantic complexity (see Section 3.3.1, Data Statistics), i.e. we collect those sentences containing only 1 predicate, up to 2 predicates, up to 3, and so on. Finally, we evaluate our models on the collected samples. Figure 3.3 confirms that the complexity of the semantic structure of a sentence is, indeed, one of the main factors behind the gap between performances of GSRL*flattened* and GSRL*nested*. This also explains why the two are much closer in CoNLL-2012, as this dataset has a significantly lower semantic complexity than CoNLL-2009 (2.8 against 4.7 predicates per sentence, respectively).

Train down-sampling: Sentence count. Even though unsupervised learning has been gaining ever more popularity in Natural Language Processing, the majority of the approaches

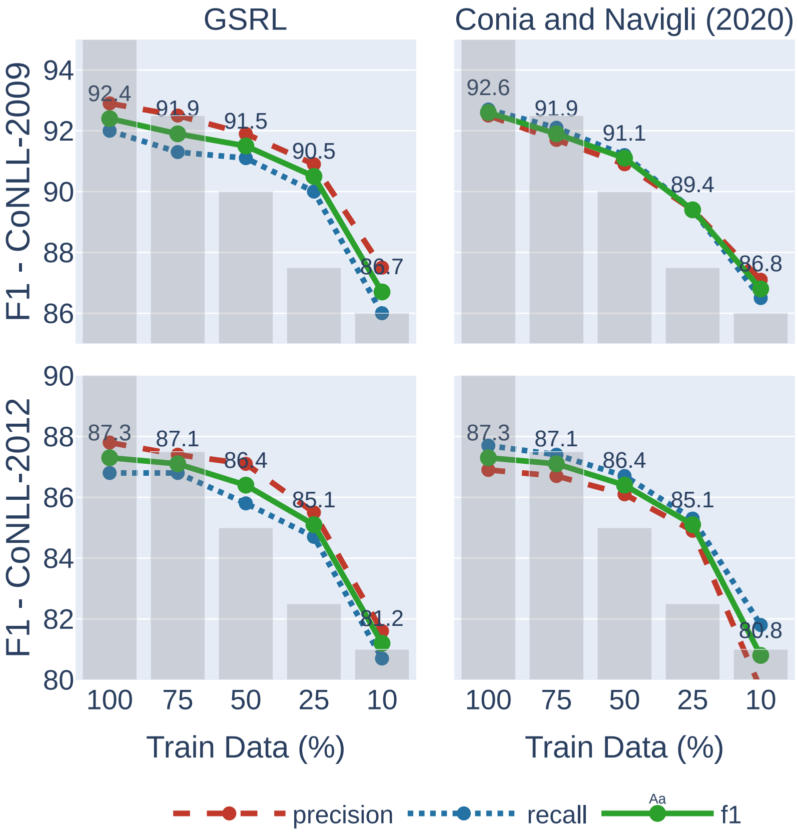


Figure 3.5. Comparison of GSRL$_{flattened}$ and Conia and Navigli (2020) system results as the train data decreases: the F$_1$ score is similar in each split (100%, 75%, 50%, 25% and 10% of the original training datasets) of both CoNLL-2009 (dependency-based) and CoNLL-2012 (span-based).

to SRL continue to rely on supervision and, therefore, on labeled data. However, the manual annotation of text with sense and role labels is an expensive process which requires money, time and expert annotators who are at ease with complex linguistic resources like PropBank, making it difficult to create large SRL datasets. In this analysis we devise a synthetic scenario in which we simulate a set of lower-resource settings and study how they affect our model. Specifically, we create different training data splits, sampling 10%, 25%, 50% and 75% of the sentences from the training data of CoNLL-2009 and CoNLL-2012 (37,847 and 90,856 sentences, respectively). As shown in Figure 3.4, when down-sampling the training data to 75% and 50% of its original size, the results decrease by less than 1.0% in F$_1$ score in the test sets of CoNLL-2009 and CoNLL-2012. On one hand, this experiment demonstrates the robustness of our model. On the other hand, it also suggests that the huge effort carried out by the creators of the CoNLL-2012 dataset to manually annotate the last

SHOT	CONLL-2009	CONLL-2012
ALL	37,847	90,856
1	5,936	4,788
2	9,227	8,085
3	11,700	10,761

Table 3.7. Number of sentences in the training samples for 1-, 2- and 3-shot learning.

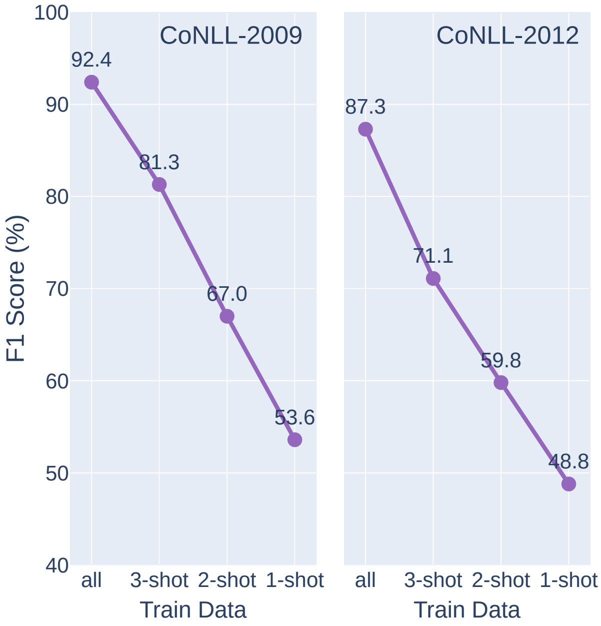


Figure 3.6. Results of GSRL$_{flattened}$ as the sentences for each predicate sense decrease: the performance goes down abruptly. This analysis shows that number of examples for each predicate sense is fundamental for a good training set.

45,000 sentences of the training set, made our model improve by only 0.9% in F_1 score. In addition, we perform the same experiment with the state-of-the-art sequence labeling system of Conia and Navigli [2020]. The side-by-side comparison is shown in Figure 3.5. Despite the drastic architectural difference between two systems, i.e., GSRL being a seq-to-seq system as opposed to the sequence labeling approach of Conia and Navigli [2020], and their different behavior in precision and recall, they converge to the same overall performance in terms of F_1 (green line) in each split on both span- and dependency-based evaluations. We argue, therefore, that simply increasing the number of training sentences is not necessarily the best direction towards better datasets and systems.

Train down-sampling: Sense count. Rather than the number of sentences in the training set, we hypothesize that a model is more susceptible to the number of times it sees a predicate sense. To test this hypothesis, we study how well GSRL is able to generalize when limiting the number of sentences for each predicate sense, i.e., how well it performs in few-shot learning. More specifically, we devise a set of three new training datasets which contain at most 1, 2 and 3 occurrences of a predicate sense by sampling the original CoNLL-2009 and CoNLL-2012 training sets. We report the sizes of these new splits in Table 3.7. Figure 3.6 shows the performance of $\text{GSRL}_{\textit{flattened}}$ in both the CoNLL-2009 and CoNLL-2012 test sets as the number of predicate sense instances in the training set decreases. While limiting the number of sentences does not result in a noteworthy impact on the results, $\text{GSRL}_{\textit{flattened}}$ shows a drastic deterioration in performance when it can only learn the predicate-argument structure of a sense from a single example (1-shot), but greatly improves when it can learn from two and three examples (2-shot and 3-shot). Not only do these results support our initial hypothesis, but they also suggest that new smaller-scale datasets, if properly devised, may still make a significant impact on a modern SRL system.

3.6 Summary

In this Chapter we presented GSRL, the first seq-to-seq model for end-to-end SRL to generate both sense and role labels. Evaluated on multiple gold benchmarks, GSRL achieves state-of-the-art results, previously attained only by sequence labeling approaches, in both span- and dependency-based English SRL. The analysis performed on our evaluation framework exposed, thanks to a set of purposely-designed synthetic scenarios, the positives and negatives of our approach, from its ability to reach competitive results with only 25% of the training data to its difficulties in modeling and generating "semantically complex" sequences. However, our analysis was not limited solely to a study of our model and, instead, we also made use of GSRL to highlight current issues, roadblocks and promising directions to further improve the area of SRL, both as regards its models and its datasets. We hope that our contributions will lead to further progress in generation-based approaches to SRL and, more importantly, open the door to their integration into more complex semantics-first tasks, such as Semantic Parsing. We release GSRL at `https://github.com/SapienzaNLP/gsrl`.

∼ *This page was intentionally left blank* ∼

Chapter 4

End-to-End AMR Parsing and Generation as Sequence Generation

Abstract

In AMR parsing, state-of-the-art parsers commonly use cumbersome pipelines integrating several different modules or components, and exploit graph recategorization, i.e., a set of content-specific heuristics that are developed on the basis of the training set. However, the generalizability of graph recategorization in an out-of-distribution setting is unclear. In contrast, AMR generation, which can be seen as the inverse to parsing, is based on simpler seq-to-seq. In this Chapter, we cast AMR parsing and AMR generation as a symmetric transduction task and show that by devising a careful graph linearization and extending a pretrained encoder-decoder model, it is possible to obtain state-of-the-art performances in both tasks using the very same seq-to-seq approach, i.e., SPRING (*Symmetric PaRsIng aNd Generation*). Our model achieves unprecedented performances, thus outperforming previous state of the art on the English AMR 2.0 benchmark by a large margin. We release the software at `https://github.com/SapienzaNLP/spring`. Finally, we make SPRING available as a service at `http://nlp.uniroma1.it/spring`.

Source: This Chapter is based on our AAAI 2021 [Bevilacqua et al., 2021a] and EMNLP 2021 [Blloshmi et al., 2021a] papers:
1. One SPRING to Rule Them Both: Symmetric AMR Semantic PaRsIng aNd Generation without a Complex Pipeline, and 2. SPRING goes Online: End-to-End AMR Parsing and Generation.

4.1 Overview

In the recent years, AMR parsing and generation models have become more reliable than they used to be, thanks to both the availability of pretrained language models [Devlin et al., 2019; Lewis et al., 2020] and the continuous improvements in the AMR-specific model architectures [Zhou et al., 2020b; Cai and Lam, 2020a; Fernandez Astudillo et al., 2020;

Mager et al., 2020]. Nevertheless, previous state-of-the-art approaches to AMR parsing featured complex pre- and postprocessing pipelines, in which the output of several different components was integrated. Additionally, they employed fine-grained, content-specific heuristics developed based on the training set that, as a consequence, could be very brittle across domains and genres. The parsing performance of simpler, full seq-to-seq methods had hitherto lagged behind, mainly because they are less data-efficient than their alternatives. In AMR generation, which can be seen as the inverse task to AMR parsing, vanilla seq2seq methods have, instead, achieved state-of-the-art results. This architectural asymmetry is not observed in other bidirectional transduction tasks such as machine translation, where the same architecture is used to handle the translation from language X to language Y, and *vice versa*.
As we showed in the previous Chapter, seq-to-seq learning can achieve state-of-the-art results even when tasked to produce semantic structures of meaning (with *proper* linearization techniques). Since SRL is a highly overlapping task with AMR, it seems natural to explore a similar direction for AMR related tasks. In this context, in our recent paper SPRING [Bevilacqua et al., 2021a], we proposed a solution to both AMR parsing and generation tasks through a simple, end-to-end approach with no heavy inbuilt data processing assumptions. Our model achieved unprecedented performance in AMR parsing and generation, both in- and out-of-distribution.

In this Chapter, we detail SPRING (*Symmetric PaRsIng aNd Generation*), a novel end-to-end approach to generating both an AMR graph when fed with an English sentence, or an English sentence when given an AMR graph as input. Our contributions are the following:

- We extend a pretrained Transformer encoder-decoder architecture to generate either an accurate linearization of the AMR graph for a sentence or, vice versa, a sentence for a linearization of the AMR graph.

- Contrary to previous reports [Konstas et al., 2017], we find that the choice between competing graph-isomorphic linearizations does matter. Our proposed Depth-First Search (DFS)-based linearization with special pointer tokens outperforms both the PENMAN linearization and an analogous Breadth-First Search (BFS)-based alternative, especially on AMR generation.

- We propose a novel OOD setting for estimating the ability of the AMR parsing and AMR generation approaches to generalize on open-world data.
- We show that rule-based graph recategorization should be avoided on open-world data because, although it slightly boosts the performance in the standard benchmark, it is not able to generalize in the OOD setting.
- We outperform the previously best reported results in AMR 2.0 by 11.2 BLEU points for the generation task, and by 3.6 Smatch points for the parsing task.

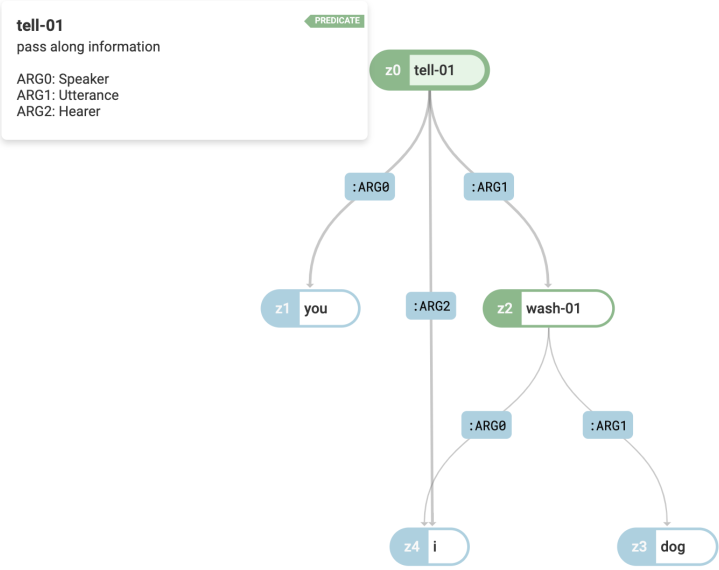
Additionally, to make SPRING accessible to the community, thereby lowering the entry point to AMR application research, we present SPRING Online Services [Blloshmi et al., 2021a] which include:

- a Web interface to easily produce and visualize an AMR graph for a given sentence and, vice versa, a sentence for a given AMR graph in PENMAN notation.
- RESTful APIs to programmatically request AMR parsing and generation services.
- a bidirectional SPRING model also trained on Bio-AMR, resulting in much stronger performances for biomedical applications.
- a feedback mechanism which allows users to submit modifications to the system's outputs – aided by the visualization – which we collect to enable future enhancements of AMR systems using active learning [Settles, 2009].

4.2 Methodology

4.2.1 Task Formulation

We perform both AMR parsing and AMR generation with the same architecture, i.e., SPRING, which exploits the transfer learning capabilities of BART for the two tasks. In SPRING AMR graphs are handled symmetrically: for AMR parsing the encoder-decoder is trained to predict a graph given a sentence; for AMR generation another specular encoder-decoder is trained to predict a sentence given a graph.



```
PM
( t / tell-01
    :ARG0 ( y / you )
    :ARG1 ( w / wash-01
        :ARG0 i
        :ARG1 ( d / dog ) )
    :ARG2 ( i / i ) )

DFS
( <R0> tell-01
    :ARG0 ( <R1> you )
    :ARG1 ( <R3> wash-01
        :ARG0 <R2>
        :ARG1 ( <R4> dog ) )
    :ARG2 ( <R2> i ) )
```

```
BFS

<R0> tell-01
    :ARG0 <R1> you
    :ARG1 <R3> wash-01
    :ARG2 <R2> i
<stop>

<R3>
    :ARG0 <R2>
    :ARG1 <R4> dog
<stop>
```

Figure 4.1. The AMR graph for the sentence “You told me to wash the dog.” with the three different linearizations.

Formally, a sentence is represented as a sequence of tokens $\mathbf{s} = \langle \mathrm{BOS}, w_1, w_2, \ldots, w_n, \mathrm{EOS} \rangle$ where each word w_i belongs to the vocabulary V, and $\mathrm{BOS}, \mathrm{EOS} \in V$ are special beginning-of-sentence and end-of-sentence tokens, respectively. For example, the sentence *You told me to wash the dog* is represented as ⟨BOS,‘You’, ‘told’, ‘me’, ‘to’, ‘wash’, ‘the’, ‘dog’, EOS⟩. Similarly, a linearized graph is also a sequence $\mathbf{g} = \langle \mathrm{BOS}, g_1, g_2, \ldots, g_m, \mathrm{EOS} \rangle$, where $g_i \in V$. The graph of the aforementioned sentence is shown in Figure 4.1. Note that both sentence and graph tokens are drawn from the same vocabulary (see Section 4.2.3).

4.2.2 Graph Linearizations

In this work we use linearization techniques which are fully graph-isomorphic, i.e., it is possible to encode the graph into a sequence of symbols and then decode it back into a graph without losing adjacency information. We propose the use of special tokens `<R0>`, `<R1>`, . . . , <Rn> to represent variables in the linearized graph and to handle co-referring nodes. Just as happens with variable names in PENMAN, i.e., the encoding that is used in the release files of AMR, whenever such special tokens occur more than once it is signaled in our encoding that a given node fulfills multiple roles in the graph. By means of this modification we aim to address the confusion arising from the use of seq-to-seq with PENMAN (PM), which does not allow a clear distinction to be made between constants and variables, as variable names have no semantics. Our special tokens approach is used in combination with two graph traversal techniques based on, respectively, DFS and BFS; in addition, we also experiment with PENMAN. In Figure 4.1 we show the linearizations of the AMR graph for "You told me to wash the dog".

DFS-based. DFS, on which PENMAN is based, is very attractive as it is quite closely related to the way natural language syntactic trees are linearized: consider, e.g., the sentence "the dog which ate the bone which my father found is sleeping", where the noun *dog* is far removed from its head verb, *is sleeping*, because the dependents of *dog* are "explored" completely before the occurrence of the head verb. Thus, we employ a DFS-based linearization with special tokens to indicate variables and parentheses to mark visit depth. Moreover, we dispose of the redundant slash token (`/`). These features significantly reduce the length of the output sequence compared to PENMAN, where variable names are often split into multiple subtokens by the subword tokenizer. This is important for efficient seq-to-seq decoding with Transformers, which are bottlenecked by the quadratic complexity of attention mechanisms.

BFS-based. The use of BFS traversal is motivated by the fact that it enforces a locality principle by which things belonging together are close to each other in the flat representation. Additionally, Cai and Lam [2019] suggest that BFS is cognitively attractive because it corresponds to a core-semantic principle which assumes that the most important pieces of meaning are represented in the upper layers of the graph. To this end, we present a

BFS-based linearization which, just like our DFS-based one, uses special tokens to represent co-reference. We apply a BFS graph traversal algorithm which starts from the graph root r and visits all the children w connected by an edge e, appending to the linearization the pointer token to r, e, and then a pointer token if w is a variable, or its value in case w is a constant. The first time a pointer token is appended, we also append its `:instance` attribute. At the end of the iteration at each level, i.e., after visiting the children w, we append a special <`stop`> token to signal the end node exploration. In Figure 4.1, the visit starts with `tell-01`, iterates over its children, then, after the <`stop`>, goes on to `wash-01`.

Edge ordering. All the above linearizations are decoded into the same graph. However, in the PENMAN-linearized gold annotations, an edge ordering can be extracted from each AMR graph. There has been a suggestion [Konstas et al., 2017] that annotators have used this possibility to encode information about argument ordering in the source sentence. Our preliminary experiments confirmed that imposing an edge ordering different from PENMAN has a big negative effect on the evaluation measures of AMR generation, due to their order-sensitive nature. To control this, we have carefully designed the linearizations to preserve order information.

Linearization information loss. Previous approaches to AMR parsing [Konstas et al., 2017; van Noord and Bos, 2017; Peng et al., 2017; Ge et al., 2019] use seq-to-seq methods in conjunction with lossy linearization techniques, which, in order to reduce complexity, remove information such as variables from the graph. This information is restored heuristically, making it harder to produce certain valid outputs. In contrast, we our proposed linearization techniques are completely isomorphic to the graph, and do not incur any information loss.

4.2.3 SPRING Model

SPRING is at its heart a function P_θ (with θ being the parameters) that takes as input a source string σ in $V^* = \bigcup_{i=1}^{\infty} V^i$ and a partial target string $\tau \in V^*$. Then P_θ outputs a next-token probability distribution over V. Applying this basic function repeatedly, we can assign a probability (P^*) to any string of tokens given another one by factorising it in a left-to-right way as a product of conditional probabilities. This can be applied both to the parsing (by using $\mathbf{s}$ as σ, and the progressively built linearization $\mathbf{g}$ as τ; Eq. 4.1) and generation (exchanging σ and τ; Equation 4.2):

$$P^*_\theta(\mathbf{g}|\mathbf{s}) = \prod_{i=1}^{m+1} P_\theta(g_i \mid \tau = \mathbf{g}_{0:i-1}, \sigma = \mathbf{s}) \tag{4.1}$$

$$P^*_\theta(\mathbf{s}|\mathbf{g}) = \prod_{i=1}^{n+1} P_\theta(s_i \mid \tau = \mathbf{s}_{0:i-1}, \sigma = \mathbf{g}) \tag{4.2}$$

To train the model we optimize the parameters to minimize, with mini-batch gradient descent, the so-called negative log likelihood $\mathcal{L}_\theta$ (the negative log conditional probability) over a dataset $\mathcal{D}$ collecting sentence-graph pairs, both for parsing ($\mathcal{L}^{\text{PAR}}_{\theta^{(1)}}$) and generation ($\mathcal{L}^{\text{GEN}}_{\theta^{(2)}}$):

$$\begin{aligned}&\underset{\theta^{(1)},\theta^{(2)}}{\text{argmin}} \quad \mathcal{L}^{\text{PAR}}_{\theta^{(1)}}(\mathcal{D}) + \mathcal{L}^{\text{GEN}}_{\theta^{(2)}}(\mathcal{D}) = \\ &\underset{\theta^{(1)},\theta^{(2)}}{\text{argmin}} \; - \sum_{\langle \mathbf{s},\mathbf{g}\rangle \in \mathcal{D}} \text{log} P^*_{\theta^{(1)}}(\mathbf{g}|\mathbf{s}) + \text{log} P^*_{\theta^{(2)}}(\mathbf{s}|\mathbf{g})\end{aligned} \tag{4.3}$$

Note that when $\theta^{(1)}$ is different from $\theta^{(2)}$, the two objective terms are optimized separately. Instead, when we enforce $\theta^{(1)} = \theta^{(2)}$ we have a model that is not only symmetric, but can also perform both AMR parsing and generation at the same time.

Once we have the trained model, the predicted output is the string ending in EOS with the highest probability in P^*_θ. Unfortunately, finding this optimal string is intractable when $|V|$ is large; in practice, however, we can perform an approximate decoding with histogram beam search.

Similarly to the GSRL model in Chapter 3 (see Section 3.2.2), SPRING is based on the Transformer architecture by Vaswani et al. [2017b], a seq-to-seq neural network that, briefly, i) uses attention instead of recurrence to encode sequences, ii) is made up of an encoder module that embeds σ, and a decoder that, based on both the encoder output and τ, produces the final distribution output. Key to the high performances of SPRING is the fact that its parameters are not randomly initialized, but, instead, are adopted from those of BART Lewis et al. [2020]. BART has shown significant improvements in conditioned generation tasks where the vocabulary of the input and output sequences largely intersect, such as question answering and summarization. Similarly, a large amount of AMR labels are drawn from the English vocabulary – despite the fact that AMR aims to abstract away from the sentence – and, therefore, we hypothesize that BART's denoising pretraining should be suitable for AMR parsing and generation as well. Moreover, it is possible to see a parallel between BART's pretraining task and AMR generation, since the linearized AMR graph can be seen as a reordered, partially corrupted version of an English sentence, which the model has to reconstruct. Owing to this, SPRING can exploit the extensive knowledge BART encompasses, gained through optimization on large amounts of raw text with an unsupervised denoising objective.

Vocabulary. BART uses a subword vocabulary and its tokenization is optimized to handle English, but it is not well-suited for AMR symbols. To deal with this problem we expand the tokenization vocabulary of BART by adding i) all the relations and frames occurring at least 5 times in the training corpus; ii) constituents of AMR tokens, such as `:op`; iii) the special tokens that are needed for the various graph linearizations. Moreover, we adjust the embedding matrices of encoder and decoder to include the new symbols by adding a vector which is initialized as the average of the subword constituents. The addition of AMR-specific symbols in vocabulary expansion avoids extensive subtoken splitting and thus allows the encoding of AMRs as a more compact sequence of symbols, cutting decoding space and time requirements.

Recategorization. Recategorization is a popular technique to shrink the vocabulary size for handling data sparsity. It simplifies the graph by removing sense nodes, wiki links, polarity attributes, and/or by anonymizing the named entities. To assess the contribution of recategorization, we experiment with a commonly-used method in AMR parsing literature [Zhang et al., 2019a,b; Zhou et al., 2020b; Cai and Lam, 2020a]. The method is based on string-matching heuristics and mappings tailored to the training data, which also regulate the restoration process at inference time. We direct the reader to Zhang et al. [2019a] for further details. We note that following common practice we use recategorization techniques only in parsing, due to the considerably higher information loss that could result in generation.

4.2.4 Postprocessing

In our approach we perform light postprocessing, mainly to ensure the validity of the graph produced in parsing. To this end, we restore parenthesis parity in PENMAN and DFS, and also remove any token which is not a possible continuation given the token that precedes it. For BFS, we recover a valid set of triples between each subsequent pair of

<stop> tokens. Our approaches remove content limited to a few tokens, often repetitions or hallucinations. We notice that non-recoverable graphs are very rare, roughly lower than 0.02% in out-of-distribution data, with a negligible effect on overall performance.[1] In addition, we integrate an external Entity Linker to handle wikification, because it is difficult to handle the edge cases with pure seq-to-seq. We use a simple string matching approach to search for a mention in the input sentence for each `:wiki` attribute that SPRING predicted in the graph, then run the off-the-shelf BLINK Entity Linker [Wu et al., 2020] and overwrite the prediction.

4.3 Experiments

4.3.1 Evaluation Benchmarks

In-Distribution. We evaluate the strength of SPRING on the standard evaluation benchmarks, which we refer to as the In-Distribution (ID) setting. The data that we use in this setting are the AMR 2.0 (LDC2017T10) and AMR 3.0 (LDC2020T02) corpora releases, which include, respectively 39,260 and 59,255 manually-created sentence-AMR pairs. AMR 3.0 is a superset of AMR 2.0. In both of them the training, development and test sets are a random split of a single dataset, therefore they are drawn from the same distribution.

Out-of-Distribution. While the ID setting enables a comparison against previous literature, it does not allow estimates to be made about performances on open-world data, which will likely come from a different distribution of that of the training set. Motivated by common practice in related semantic tasks, such as Semantic Role Labeling [Hajič et al., 2009], we propose a novel OOD setting.

In this evaluation setting we assess the performance of SPRING when trained on OOD data, contrasting it with the ID results. We employ the AMR 2.0 training set, while for testing we use three distinct Out-of-Distribution (OOD) benchmarks, covering a variety of different genres:

[1] It might be interesting to see whether the tree decoding approaches presented in recent work by Prange et al. [2021] could be employed to avoid invalid graph generation as future work.

i) **New3**, a set of 527 instances from AMR 3.0, whose original source was the LORELEI DARPA project – not included in the AMR 2.0 training set – consisting of excerpts from newswire and online forums;

ii) **TLP**, the full AMR-tagged children's novel *The Little Prince* (ver. 3.0), consisting of 1,562 pairs;

iii) **Bio**, i.e., the test set of the Bio-AMR corpus, consisting of 500 instances, featuring biomedical texts [May and Priyadarshi, 2017].

Silver dataset. In order to determine whether silver-data augmentation, another commonly used technique, is beneficial in both ID and OOD, we follow Konstas et al. [2017] and create pretraining data by running the SPRING parser using DFS (trained on AMR 2.0) on a random sample of the Gigaword (LDC2011T07) corpus consisting of 200,000 sentences.

4.3.2 Evaluation Metrics

We evaluate on the AMR parsing benchmarks by using Smatch [Cai and Knight, 2013] computed with the tools released by Damonte et al. [2017], which also report fine-grained scores on different aspects of parsing, such as wikification, concept identification, NER and negations. As regards AMR generation, we follow previous approaches and evaluate using three common Natural Language Generation (NLG) measures, i.e., BLEU [Papineni et al., 2002, BL], chrF++ [Popović, 2017, CH+], and METEOR [Banerjee and Lavie, 2005, MET], tokenizing with the script provided with JAMR [Flanigan et al., 2014]. Additionally, as AMR abstracts away from many lexical and syntactic choices, we report the scores with untokenized BLEURT [Sellam et al., 2020, BLRT], i.e., a recent regression-based measure which has shown the highest correlation with human judgements in machine translation.

4.3.3 Training and Tuning

SPRING relies on BART with the augmented vocabulary, as discussed in Section 4.2.3. We use the same model hyperparameters as BART Large (or Base, when specified), as defined in Huggingface's `transformers` library. Models are trained for 30 epochs using cross-entropy with a batch size of 500 graph linearization tokens, with RAdam [Liu et al.,

PARAMETER	PICK	SEARCH SPACE
Training		
Learning Rate (LR)	$5 * 10^{-5}$	1/5/10/50 $*10^{-5}$
LR Scheduling	constant	-
Betas	0.9, 0.999	-
Dropout	0.25	0.1 to 0.25, (+0.05)
Weight Decay	0.004	0.001 to 0.01, (+0.001)
Gradient Accumulation	10	1/5/10/15/20
Prediction		
Beam size	5	[1,5]

Table 4.1. Hyperparameters and search space.

2020b] optimizer and a learning rate of 1×10^{-5}. Gradient is accumulated for 10 batches. Dropout is set to 0.25.

Hyperparameter search. We report in Table 4.1 the final hyperparameters used to train and evaluate both the AMR parsing and AMR generation models. To pick these parameters, we used random search with about 25 AMR parsing trials in the search space indicated in the third column. AMR parsing training requires about 22 and 30 hours on AMR 2.0 and AMR 3.0 using one 1080 Ti GPU, respectively; AMR generation requires 13 and 16.5 hours on AMR 2.0 and AMR 3.0, respectively. At prediction time, we set beam size to 5 following common practice in neural machine translation [Yang et al., 2018].

SPRING variants. We include models trained with the three linearizations, indicated as SPRING$^{\text{[lin]}}$, where [lin] is one of the linearizations: PENMAN (PM), DFS- (DFS) or BFS-based (BFS). In addition, we include variants of SPRING$^{\text{DFS}}$ using i) BART Base (base); ii) graph recategorization (+recat); iii) pretrained silver AMR data (+silver). We also report results on a vanilla BART baseline which treats PENMAN as a string, uses no vocabulary expansion and tokenizes the graph accordingly.

4.3.4 Comparison Systems

In-Distribution. In the ID setting, we use the AMR 2.0 benchmark to compare SPRING variants against the best models from the literature. To this end, we include the following AMR parsers:

i) Ge et al. [2019, Ge+], an encoder-decoder model which encodes the dependency tree and semantic role structure alongside the sentence;

ii) Lindemann et al. [2019, LindGK], a compositional parser based on the *Apply-Modify* algebra;

iii) Naseem et al. [2019, Nas+], a transition-based parser trained with a reinforcement-learning objective rewarding the Smatch score;

iv) Zhang et al. [2019b, Zhang+], a hybrid graph- and transition-based approach incrementally predicting an AMR graph;

v) Zhou et al. [2020b, Zhou+], an aligner-free parser [Zhang et al., 2019a] enhanced with latent syntactic structure;

vi) Cai and Lam [2020a, CaiL], a graph-based parser iteratively refining an incrementally constructed graph.

For AMR generation, instead, we include the following:

i) Zhu et al. [2019b, Zhu+], a Transformer-based approach enhanced with structure-aware self-attention;

ii) Cai and Lam [2020b, CaiL], a graph Transformer model which relies on multi-head attention [Vaswani et al., 2017c] to encode an AMR graph in a set of node representations;

iii) Wang et al. [2020, Wang+], a Transformer-based model generating sentences with an additional structure reconstruction objective;

iv) Zhao et al. [2020, Zhao+], a graph attention network which explicitly exploits relations by constructing a line graph;

v) Yao et al. [2020, Yao+], a graph Transformer-based model which encodes heterogeneous subgraph representations;

vi) Mager et al. [2020, Mag+], a fine-tuned GPT-2 model [Radford et al., 2019] predicting the PENMAN linearization of an AMR graph.

Model	Recat.	Smatch	Unlab.	NoWSD	Conc.	Wiki.	NER	Reent.	Neg.	SRL
Ge+ (2019)	N	74.3	77.3	74.8	84.2	71.3	82.4	58.3	64.0	70.4
LindGK ((2019)**	N	75.3	-	-	-	-	-	-	-	-
Nas+ ((2019)**	N	75.5	80.0	76.0	86.0	80.0	83.0	56.0	67.0	72.0
Zhang+ ((2019b)**	Y	77.0	80.0	78.0	86.0	86.0	79.0	61.0	77.0	71.0
Zhou+ ((2020b)*	Y	77.5	80.4	78.2	85.9	86.5	78.8	61.1	76.1	71.0
CaiL ((2020a)*	N	78.7	81.5	79.2	88.1	81.3	87.1	63.8	66.1	74.5
CaiL ((2020a)*	Y	80.2	82.8	80.0	88.1	86.3	81.1	64.6	78.9	74.2
SPRING^{DFS}	N	83.8	86.1	84.4	90.2	84.3	**90.6**	70.8	74.4	79.6
SPRING^{BFS}	N	83.2	85.7	83.7	90.3	83.5	90.2	70.9	70.9	78.2
SPRING^{PM}	N	83.6	86.1	84.1	90.1	83.1	90.2	71.4	72.7	79.4
BART baseline	N	82.7	85.1	83.3	89.7	82.2	90.0	70.8	72.0	79.1
SPRING^{DFS} (base)	N	82.8	85.3	83.3	89.6	83.5	89.9	70.2	71.5	79.0
SPRING^{DFS} +recat	Y	**84.5**	**86.7**	**84.9**	89.6	**87.3**	83.7	72.3	**79.9**	79.7
SPRING^{DFS} +silver	N	84.3	**86.7**	84.8	**90.8**	83.1	90.5	**72.4**	73.6	**80.5**

Table 4.2. AMR parsing results (AMR 2.0). Row blocks: previous approaches; SPRING variants; baseline + other SPRING^{DFS}. Columns: model; recategorization (Y/N); Smatch; Fine-grained scores. The best result per measure across the table is shown in **bold**. The best result per measure within each row block is underlined. Models marked with */** rely on BERT Base/Large.

For AMR 3.0, which is a recent benchmark, there are no previous systems to compare against. Thus, we train the previous state-of-the-art parsing model of Cai and Lam [2020a] on AMR 3.0 and perform the corresponding evaluation.

Out-of-Distribution. In the OOD setting we compare the SPRING^{DFS} variants when trained on AMR 2.0 and test on OOD data (New3, Bio and TLP) against the best of the same variants trained on the corresponding ID training set when available (i.e., New3 and Bio).

4.4 Results

We now report the results of our experiments. First, we evaluate SPRING on AMR 2.0 parsing and generation; then, we show, for the first time, the figures on the new AMR 3.0 benchmark. Finally, we tackle our proposed OOD setting.

4.4.1 AMR 2.0 Results

AMR parsing. The results on the AMR 2.0 benchmark are reported in Table 4.2. Among the three different simple linearization models, i.e., SPRING^{DFS}, SPRING^{BFS}, and SPRING^{PM}, the DFS-based one achieves the highest overall Smatch, obtaining slightly better results than the second-best one, the PENMAN, and a wider margin over the BFS one. All our configurations, however, outperform previous approaches by a large margin, with SPRING$^{\text{DFS}}$ outscoring the recategorized model of Cai and Lam [2020a] by 3.6 F1 points. The score gains are spread over most of the fine-grained categories of Damonte et al. [2017], shown in the third column block in Table 4.2. The only notable exceptions are wikification and negations, where the score of SPRING$^{\text{DFS}}$ is lower than that of the previous state of the art, i.e., Cai and Lam [2020a], which handles both wiki links and negations heuristically. When we use recategorization, i.e., in SPRING$^{\text{DFS}}$+recat, we obtain a significant boost in performance, which is especially notable in the two above-mentioned categories. Moreover, SPRING$^{\text{DFS}}$+recat achieves the best reported overall performance so far, i.e., 84.5 Smatch F1 points. Regarding the other variants of SPRING$^{\text{DFS}}$, we inspect the contribution of silver data pretraning, i.e., SPRING$^{\text{DFS}}$+silver, and notice a significant improvement over SPRING$^{\text{DFS}}$, suggesting that warm-starting the learning is beneficial in this setting. Indeed, the model of Ge et al. [2019], which does not exploit pretraining, performs considerably worse. We note, however, that in addition to the powerful initialization of BART, our extensions also provide a significant improvement over the BART baseline, ranging from 0.5 (SPRING$^{\text{BFS}}$) to 1.1 (SPRING$^{\text{DFS}}$) Smatch points. Finally, even when we limit the number of parameters, and use BART Base instead, we outperform the previous state of the art, obtaining 82.8 Smatch F1 points.

Finally, we compute the significance of performance differences among SPRING variants using the non-parametric approximate randomization test [Riezler and Maxwell, 2005], which is very conservative and appropriate for corpus-level measures. The improvement of SPRING$^{\text{DFS}}$ against SPRING$^{\text{BFS}}$ and BART baseline is significant with $p < 0.005$, while it is not significant when considering PENMAN linearization.

AMR generation. We report in Table 4.3 the AMR 2.0 AMR generation results. SPRING$^{\text{DFS}}$ achieves 45.3 BLEU points, improving the previous state of the art [Yao et al., 2020] by 11 points, and obtains very significant gains in chrF++ and METEOR as well. As far as linearization is concerned, SPRING$^{\text{DFS}}$ proves to be significantly stronger than both SPRING$^{\text{PM}}$ and SPRING$^{\text{BFS}}$ in 3 out of the 4 measures.

This could be due to the fact that DFS is closer to natural language than BFS, and is more

	BL	CH+	MET	BLRT
Zhu+ (2019b)	31.8	64.1	36.4	-
CaiL (2020b)	29.8	59.4	35.1	-
Wang+ (2020)	32.1	64.0	36.1	-
Zhao+ (2020)	32.5	-	36.8	-
Mag+ (2020)	33.0	63.9	37.7	-
Yao+ (2020)	34.1	65.6	38.1	-
SPRING$^{\text{DFS}}$	45.3	73.5	41.0	56.5
SPRING$^{\text{BFS}}$	43.6	72.1	40.5	54.6
SPRING$^{\text{PM}}$	43.7	72.5	41.3	56.0
BART baseline	42.7	72.2	40.7	54.8
SPRING$^{\text{DFS}}$ +silver	**45.9**	**74.2**	**41.8**	**58.1**

Table 4.3. AMR generation results (AMR 2.0). Row blocks: previous approaches; SPRING variants; baseline +silver. Columns: measures. **Bold**/underline as in Table 4.2.

compact and efficient than PENMAN (see Section 4.2.2). Similarly to the AMR parsing task results, the pretraining with silver data boosts the performance, with SPRING$^{\text{DFS}}$+silver improving the baseline by 0.6 BLEU points. Finally, there is a big gain against the fine-tuned GPT-2 model of Mager et al. [2020], demonstrating that using a pretrained decoder on its own is suboptimal. As in AMR parsing, we compute the significance of results using the non-parametric approximate randomization test. The performance gap between SPRING$^{\text{DFS}}$ and the alternatives in AMR generation, i.e., SPRING$^{\text{PM}}$, SPRING$^{\text{BFS}}$, and BART baseline, is significant with $p < 0.001$.

4.4.2 AMR 3.0 Results

The results on AMR 3.0 (Table 4.4) confirm that SPRING$^{\text{DFS}}$ obtains the best performance. However, the important thing to note here is that graph recategorization, without significant human effort in expanding the heuristics,[2] is not able to scale on a more diverse benchmark such as AMR 3.0: SPRING$^{\text{DFS}}$+recat achieves lower performances than the non-recategorized counterpart, with the exception of negations, whose heuristics are probably more resilient to change in data distribution. Note that the harmful impact of recategorization outside of AMR 2.0 is noticeable even with the pretrained model of Cai and Lam [2020a].

[2] We use the heuristics designed by Zhang et al. [2019a] which were optimized on the AMR 2.0 training set.

	CaiL (2020a)	CaiL (2020a)+recat	SPRING$^{\text{DFS}}$	SPRING$^{\text{DFS}}$+silver	SPRING$^{\text{DFS}}$+recat
AMR Parsing					
Smatch	78.0	76.7	**83.0**	**83.0**	80.2
Unlab.	81.9	80.6	**85.4**	**85.4**	83.1
NoWSD	78.5	77.2	**83.5**	**83.5**	80.7
Conc.	88.5	86.5	**89.8**	89.5	87.7
Wiki.	75.7	77.3	**82.7**	81.2	77.8
NER	83.7	74.7	**87.2**	87.1	79.8
Reent.	63.7	62.6	70.4	**71.3**	69.7
Neg.	68.9	72.6	73.0	71.7	**75.1**
SRL	73.2	72.2	78.9	**79.1**	78.1
AMR Generation					
BL	-	-	44.9	**46.5**	-
CH+	-	-	72.9	**73.9**	-
MET	-	-	40.6	**41.7**	-
BLRT	-	-	57.3	**60.8**	-

Table 4.4. AMR parsing and AMR generation results on AMR 3.0. Best in **bold**. S$^{\text{[lin]}}$ = SPRING$^{\text{[lin]}}$. +s/r = +silver/recat.

	New3	**TLP**	**Bio**
AMR Parsing			
SPRING$^{\text{DFS}}$ (ID)	78.6	-	79.9
SPRING$^{\text{DFS}}$	**73.7**	77.3	**59.7**
SPRING$^{\text{DFS}}$+recat	63.8	76.2	49.5
SPRING$^{\text{DFS}}$+silver	71.8	**77.5**	59.5
AMR Generation			
SPRING$^{\text{DFS}}$ (ID)	61.5	-	32.3
SPRING$^{\text{DFS}}$	**51.7**	**41.5**	5.2
SPRING$^{\text{DFS}}$+silver	50.2	40.4	**5.9**

Table 4.5. OOD evaluation on AMR parsing (Smatch) and AMR generation (BLEURT). Best in **bold**.

4.4.3 Out-of-Distribution Results

Finally, we show in Table 4.5 the results of the evaluation on the OOD datasets. As can be seen, there is constantly a big difference between the score achieved by the OOD models and the best ID counterparts (see OOD paragraph in Section 4.3.4), indicated as SPRING$^{\text{DFS}}$ (ID). Interestingly enough, not using recategorization results in consistently higher performances than using it. This is especially notable for Bio, which, in addition to being OOD with respect to the AMR 2.0 training set, is also out-of-domain. On this dataset SPRING$^{\text{DFS}}$ (ID) model outperforms SPRING$^{\text{DFS}}$ by over 20 Smatch points, and SPRING$^{\text{DFS}}$+recat by

SPRING$^{\text{DFS}}$	SPRING$^{\text{DFS}}$+recat
(1) *I didn't say he believes that.*	

(s / say-01
  :polarity -
  :ARG0 (i / i)
  :ARG1 (b / believe-01
    :ARG0 (h / he)
    :ARG1 (t / that)))
	(s / say-01
	  :polarity -
	  :ARG0 (i / i)
	  :ARG1 (b / believe-01
	    :ARG0 (h / he)
	    :ARG1 (t / that)))

(2) *I didn't say he said that.*	

(s / say-01
  :polarity -
  :ARG0 (i / i)
  :ARG1 (s2 / say-01
 	
    :ARG0 (h / he)
    :ARG1 (t / that)))
	(s / say-01
	
	  :ARG0 (i / i)
	  :ARG1 (s2 / say-01
	    :polarity -
	    :ARG0 (h / he)
	    :ARG1 (t / that)))

(3) *Don't eat or drink*	

(o / or
  :op1 (e / eat-01
    :mode imperative
    :polarity -
    :ARG0 (y / you))
  :op2 (d / drink-01
    :mode imperative
    :polarity -
    :ARG0 y))
	(o / or
	  :op1 (e / eat-01
	    :mode imperative
	    :polarity -
	    :ARG0 (y / you))
	  :op2 (d / drink-01
	    :mode imperative
	
	    :ARG0 y))

Table 4.6. Negation examples.

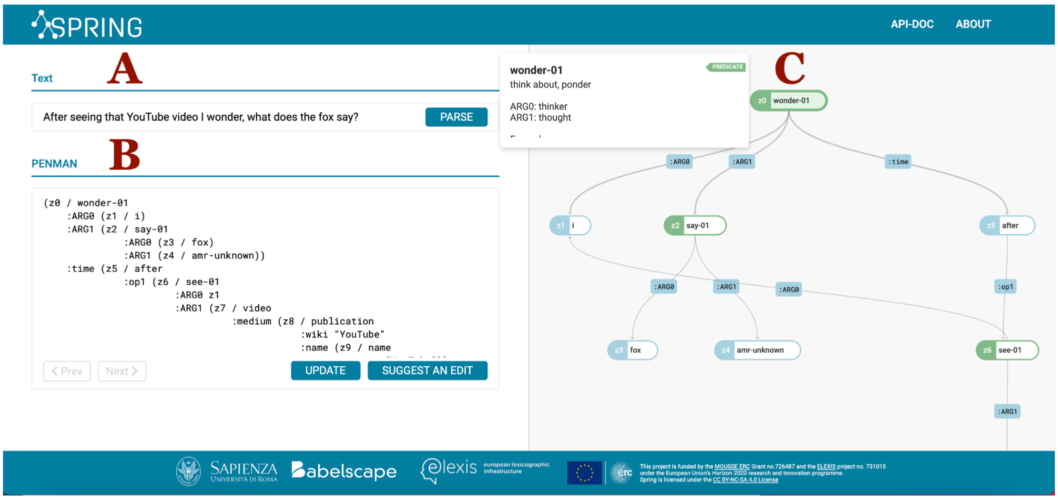
over 30 points. On New3, which is not out-of-domain, the difference with ID is noticeably narrower compared to SPRING$^{\text{DFS}}$ (4.9 Smatch points), but considerably larger against the SPRING$^{\text{DFS}}$+recat. Recategorization is not as harmful in TLP, perhaps because the text of the underlying children's story is simpler. Differently from the results on AMR 2.0, SPRING$^{\text{DFS}}$ +silver does not show consistent improvements over SPRING$^{\text{DFS}}$. We attribute this to the fact that the pretraining corpus, i.e., Gigaword, is similar in distribution to AMR 2.0, so that the boost in performance in AMR 2.0 benchmark comes due to overfitting on some genres and is not general.

4.5 Analysis

Through the OOD and AMR 3.0 benchmark evaluation, we demonstrated the harmful impact of recategorization rules based on training sets. Interestingly, across experiments, the breakdown scores [Damonte et al., 2017] for many aspects of meaning were consistently better without recategorization, with the exception of negations. Negations are handled by a commonly-used rule-based method [Zhang et al., 2019a]: `:polarity` attributes are discarded during training – causing a loss of information – and are restored by i) identifying the negated lemmas usually associated with negative polarity words such as *no, not* and *never*; ii) aligning the lemma to the corresponding node in the graph by string-matching heuristics; iii) adding the `:polarity` attribute to the aligned node. Hand-crafted rules lead to high precision due to the frequency of common patterns. However, there are many cases which the heuristics cannot handle correctly, while fully-learned approaches are able to, as they do not constrain the possible outputs they produce. In Table 4.6 we contrast the predictions of SPRING$^{\text{DFS}}$ with SPRING$^{\text{DFS}}$ +recat, trained on AMR 2.0, on several edge cases which heuristics fail to handle. Example (1) shows a standard negation with *don't* + verb, which the designed heuristics handle easily. However, simply changing a word, as in example (2), makes the rule-based system crucially depend on word-to-node alignment, which is non-trivial when the same lemma (*say*) appears multiple times. Thus, in this case, the heuristics misalign the negated occurrence of *say*, and introduce `:polarity` at a lower level in the graph. Additionally, syntax makes it such that assumptions based on word order may easily fail. However, even if the heuristics were rewritten to take syntax into account, it would still be difficult to handle cases like example (3): the negation *don't* takes large scope over the conjunction, resulting in many `:polarity` edges in the AMR graph. Finally, while due to space constraints the analysis here is limited to negations, similar problems tend to appear whenever fine-grained rules are applied to the input sentence, e.g., for entities, dates or politeness markers.

4.6 SPRING Online Services

Differently from SPRING models shown in the previous Sections, for the online services we train separate models that differ in that i) SPRING demonstration models are trained in the concatenation of AMR 3.0 and BioAMR corpus to increase generalizability, and ii) SPRING demonstration models for parsing and generation share the same learnable parameters, i.e., we train one single model to perform both tasks (refer to Equation 4.3). We provide detailed analysis of how we evaluate the models included in SPRING Online



After seeing that YouTube video I wonder, what does the fox say?

```
(z0 / wonder-01
    :ARG0 (z1 / i)
    :ARG1 (z2 / say-01
              :ARG0 (z3 / fox)
              :ARG1 (z4 / amr-unknown))
    :time (z5 / after
              :op1 (z6 / see-01
                       :ARG0 z1
                       :ARG1 (z7 / video
                                 :medium (z8 / publication
                                             :wiki "YouTube"
                                             :name (z9 / name
```

wonder-01
think about, ponder

ARG0: thinker
ARG1: thought

This project is funded by the MOUSSE ERC Grant no.726487 and the ELEXIS project no. 731015 under the European Union's Horizon 2020 research and innovation programme.
Spring is licensed under the CC BY-NC-SA 4.0 License

Figure 4.2. User interface of the SPRING *parser* Results View when the English sentence "After seeing that YouTube video I wonder, what does the fox say?" is typed as input.

Services in Appendix A.1.

In what follows we describe the functionalities of the Web interface (Section 4.6.1) and those of the RESTful APIs (Section 4.6.2) through which we make SPRING available to the community.

4.6.1 Web Interface

The main functionalities of the Web interface include switching between *parsing* and *generation* modalities, visual inspection of SPRING results view and the feedback mechanism we develop to enable users to validate SPRING predictions.

The modality can be set on the initial homepage by choosing `Text` or `PENMAN` from the Tab menu, with `Text` being the default option. When the `Text` option is chosen, the user is required to provide a plaintext sentence and they will then be redirected to the SPRING *parser* Results View (shown in Figure 4.2). On the other hand, when the `PENMAN` option is chosen, the user is required to type or copy a valid AMR graph in PENMAN notation. In the case when the PENMAN provided is valid, the user is redirected to the SPRING *generator* Results View. Otherwise, when the graph is not valid, the user is notified by a warning which points to the error line number of the PENMAN.

The Results View is similar for both parsing and generation, and we only exchange the query (input) box and the result (output) box. It consists of the following components;

A. **Query box**: As in the Modality Selector phase, also here, in the *parsing* modality the query box takes as input a plaintext sentence as input, while in *generation* the query box requires the input to be a valid PENMAN. A user can parse or generate from different inputs in this view while remaining in the same modality. To switch from *parsing* to *generation* or vice versa, the user should go back to the initial homepage.

B. **Result box**: When parsing a sentence, the Result box will be filled with the predicted graph in PENMAN format. This box is editable to enable user feedback. When *generating* from an AMR graph, the Result box shows the generated sentence which can also be modified by the user and submitted to the feedback system.

C. **AMR view panel**: This is a key component of the Results View, which visualizes an AMR as a hierarchical graph with labeled nodes and labeled edges. We devise a custom node and edge layout meant to enhance readability even in the case of big graphs with a lot of coreference edges. For example, there might be overlapping edges, edge labels or nodes in the graph. To increase visibility, the user can click/hover on an edge or edge label, and it will be highlighted and brought to the foreground. The same applies to nodes, and in addition, clicking/hovering over nodes will also highlight and bring to the foreground every incoming and outgoing edge, thus identifying all the local relations of a concept. The graph view is resizeable in order to better handle big AMR graphs, and the user is also able to zoom in/out for ease of reading. There are 4 types of node, indicated by different colors, comprising: i) predicate concept nodes, ii) non-predicate concept nodes, iii) constant nodes and iv) wiki nodes. Both predicate and non-predicate nodes are labeled with a variable name and the concept they represent. The variable makes it easy to locate the node in the PENMAN box on the left Panel.

Futhermore, both predicate and wiki nodes are associated with an `onhover/onclick` tooltip box that further defines them. The tooltip associated with the wiki node contains information taken from the corresponding BabelNet[3] Navigli and Ponzetto [2010]; Navigli et al. [2021] concept, displaying a short entity description and image (when applicable), also redirecting the user to the corresponding BabelNet page when clicking on it. This choice is motivated by the fact that BabelNet concepts function as a hub of

[3] Version 5.0.

information beyond that of Wikipedia, which paves the way for future integration of other resources in AMR. The tooltip of the predicate node, instead, provides details on the predicate definition and arguments taken from the PropBank framesets Palmer et al. [2005]. In addition, we display an example sentence containing the predicate in the specified sense. The user is redirected to the PropBank predicate page when clicking the tooltip. We mean the extra information shown by the tooltip component to be useful for the user to identify potential parsing mistakes in the output of the system, and ideally to use the provided feedback mechanism to suggest corrections.

Finally, one key functionality of SPRING Online Services that requires user interaction is the Feedback Mechanism. It is included in both parsing and generation modalities. With this feature, we aim to obtain a manual validation of SPRING output graphs or sentences, aided by the visualization. More specifically, when a user recognizes a mistake of the SPRING *parser*, including both missing or extra nodes and edges, or wrongly labeled ones, they are allowed to suggest modifications. In SPRING *parser* modality, multiple modifications are allowed in the left-panel PENMAN box, which are updated simultaneously in the right AMR view panel when the `UPDATE` button is pressed, and a user can then navigate through their own modifications by means of the `Prev` and `Next` buttons. To submit a final modification request, a user is provided with the `SUGGEST AN EDIT` button. The modifications are accepted if they lead to a correctly-formed graph. When this is the case, we save the modification request in a database for further validation. In contrast, when a mistake is found the user is warned about the line in PENMAN where it occurs. In the SPRING *generator* instead, only the predicted sentence is allowed to be modified, assuming that the input graph by the user is correct and does not need further modification. If this is not the case, the user can query the system with another AMR to obtain a new result. This feedback mechanism paves the way to future advancements in the field:

- enabling the use of active learning for improving system performance;
- collecting human validated SPRING output which can be further used as synthetic data for enhancing AMR systems;
- providing evidence of common SPRING mistakes which can aid studies on interpretation and reinforcement of AMR systems' knowledge.

Since data collection requires time and considerable interaction of users with our services, we leave the exploration of methods for including such data in AMR tasks as future work. Moreover, we plan to release the accumulated data periodically and on-request to the community.

4.6.2 RESTful APIs

The RESTful APIs we provide can be used effectively to query the SPRING services programmatically. Our APIs are simple and, differently from our Web interface, do not allow modification requests of the SPRING output. The APIs can be accessed through GET or POST requests. In fact, the APIs consist of two endpoints, namely, `/api/text-to-amr` and `/api/amr-to-text`, to parse into or generate from an AMR graph, respectively. The former requires a `sentence` string parameter and the output is a JSON object containing the PENMAN graph, while the latter expects a valid string serialized PENMAN graph, and the response is a JSON object containing the sentence. To ease the usage of the RESTful APIs, the full documentation is accessible through the SPRING Web interface, i.e., `API-Doc` from the header menu bar.

4.7 Summary

In this Chapter we presented a simple, symmetric approach for performing state-of-the-art AMR parsing and AMR generation with a single seq-to-seq architecture. To achieve this, we extend a Transfomer encoder-decoder model pretrained on English text denoising to also work with AMR. Furthermore, we put forward a novel AMR graph DFS-based linearization which, in addition to being more compact than its alternatives, does not incur any information loss. Most importantly, we drop most of the requirements of competing approaches: cumbersome pipelines, heavy heuristics (often tailored to the training data), along with most external components. Despite such cutting down on complexity, we strongly outperform the previous state of the art on both parsing and generation, reaching 83.8 Smatch and 45.3 BLEU, respectively. We also propose an Out-of-Distribution setting, which enables evaluation on different genres and domains from those of the training set. Thanks to this setting, we are able to show that the integration of recategorization techniques or silver data – popular techniques for boosting performances – harm the performances in both parsing and generation. Employing a simpler approach like ours, based on lighter assumptions, allows for more robust generalization. Here we show the generalizability of the models on different data distributions and across domains, while leaving the extension across languages as in Blloshmi et al. [2020] for future work. Finally, we invite the community to use the OOD evaluation to enable the development of more robust automatic AMR approaches. Furthermore, we believe our contributions will open up more directions towards the integration of parsing and generation. To this end, we also make available SPRING Online Services, with which we bring state-of-the-art AMR systems into the hands of the community, providing a highly interactive interface and easily integrable APIs. We release our software at `https://github.com/SapienzaNLP/spring` and SPRING Online Services at `http://nlp.uniroma1.it/spring`.

∼ This page was intentionally left blank ∼

Chapter 5

AMR as an Interlingua

Abstract

Abstract Meaning Representation is agnostic about how to derive meanings from strings and for this reason it lends itself well to the encoding of semantics across languages. However, cross-lingual AMR parsing is a hard task, because training data are scarce in languages other than English and the existing English AMR parsers are not directly suited to being used in a cross-lingual setting. In this Chapter we tackle these two problems so as to enable cross-lingual AMR parsing: we explore different transfer learning techniques for producing automatic AMR annotations across languages and develop a cross-lingual AMR parser, XL-AMR. This can be trained on the produced data and does not rely on AMR aligners or *source-copy* mechanisms as is commonly the case in English AMR parsing. The results of XL-AMR significantly surpass those previously reported in Chinese, German, Italian and Spanish. Finally we provide a qualitative analysis which sheds light on the suitability of AMR across languages. We release XL-AMR at `https://github.com/SapienzaNLP/xl-amr`.

Source: This Chapter is based on our EMNLP 2020 paper [Blloshmi et al., 2020]:
XL-AMR: Enabling Cross-Lingual AMR Parsing with Transfer Learning Techniques.

5.1 Overview

Due to its flexibility, Abstract Meaning Representation (AMR) started gaining popularity not only in English but also in other languages. However, AMR was initially designed for encoding the meaning of English sentences and made extensive use of PropBank, which is not available in many languages. In addition to that, the available resources and modeling techniques focused mainly on English while leaving cross-lingual abilities of AMR parsing understudied. Damonte and Cohen [2018] proposed the task of cross-lingual AMR parsing, which uses English-centric AMR as an interlingua, i.e., to represent parallel or comparable sentences across languages using the same AMR structure, where nodes are either English words, PropBank framesets or special AMR keywords. Even after this proposal, cross-lingual AMR *parsing* received relatively less attention. This lack of interest could be mainly attributable to the lack of training data and evaluation benchmarks in languages other than English. At the time, Damonte and Cohen [2018] put forward the only cross-lingual parser and, two years later, they released a cross-lingual AMR evaluation benchmark [Damonte and Cohen, 2020]. The authors adapted a transition-based English AMR parser [Damonte et al., 2017] for cross-lingual AMR parsing, which relied on word-to-word and word-to-node automatic alignments and was trained on silver annotated data. Nevertheless, the performances it achieved were not satisfying in terms of Smatch score [Cai and Knight, 2013], mostly as a result of concept identification errors, which in turn were directly related to the usage of noisy word-to-node alignments projected from English. In this context, to address the gaps in cross-lingual AMR research, we presented XL-AMR [Blloshmi et al., 2020]. Furthermore, owing to the large success of the sequence-to-graph transduction models at the time, XL-AMR follows the same learning paradigm as the state-of-the-art models at the time of writing [Zhang et al., 2019a,b].

In this Chapter, we detail XL-AMR, a cross-lingual AMR parser aided by different transfer learning techniques: i) model transfer which relies on language-independent features, ii) annotation projection relying on parallel corpora and available English AMR parsers, and iii) automatic translation of the training corpora which guarantees gold AMR structures. The contributions of this work are:

- Development and release of XL-AMR, a cross-lingual AMR parser which disposes of word aligners, i.e., word-to-word and word-to-node, and surpasses the previously reported results on Chinese, German, Italian and Spanish, by a large margin.
- Exploration of different techniques to create cross-lingual AMR training data, showing that it is possible to transfer semantic structure information across different languages.
- Creation and release of diverse quality silver data for cross-lingual AMR parsing.
- Qualitative analysis of the ability of XL-AMR to transfer semantic structures across languages and of AMR to represent the meaning of sentences cross-lingually.

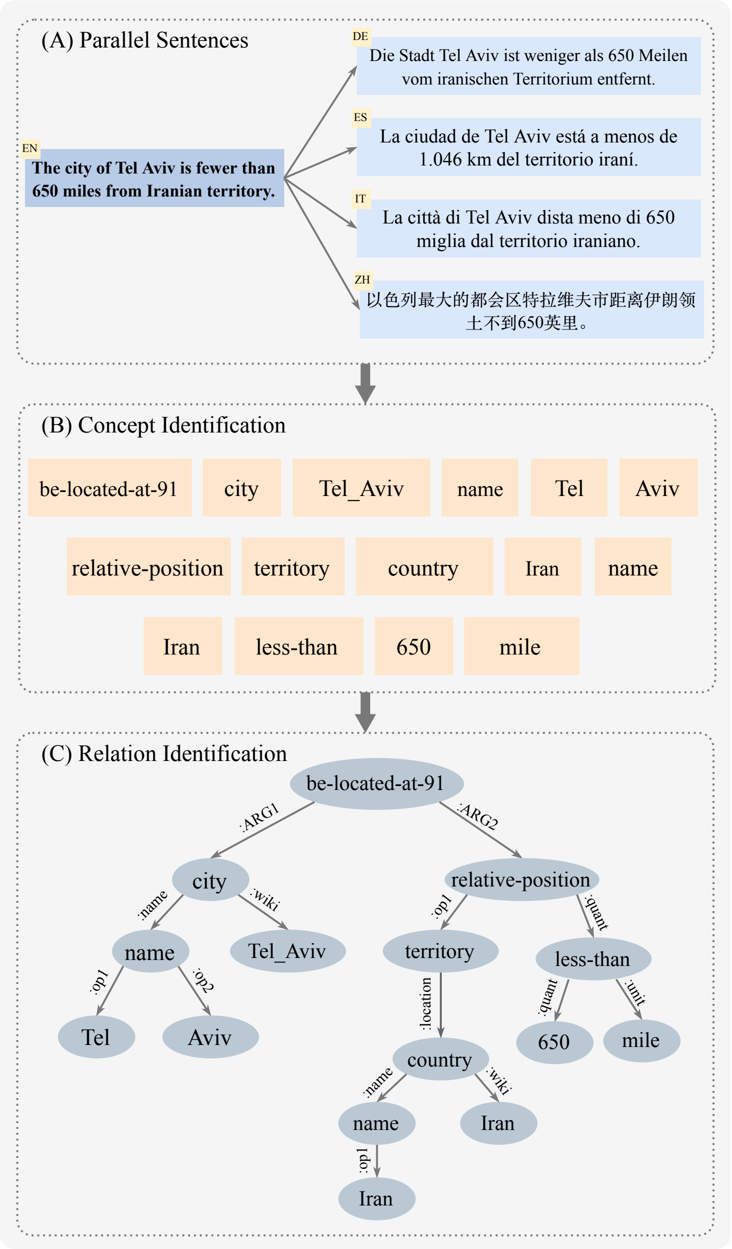
5.2 Methodology

In what follows we first formalize the task (Section 5.2.1) and then detail our cross-lingual AMR parser (Section 5.2.2) and our proposed silver data creation methods (Section 5.2.3). Finally, we list the pre- and postprocessing cross-lingual techniques and resources we employ (Section 5.2.4).

5.2.1 Cross-lingual AMR

Cross-lingual AMR parsing is defined as the task of transducing a sentence in any language to the AMR graph of its English translation whose nodes are either English words, PropBank framesets [Kingsbury and Palmer, 2002] or special AMR keywords.

Breaking down this definition, given an English sentence and its translation T_L in a language L, their meaning representation is ideally formalized by the same AMR, $G = (V, E)$, where V is a list of concept nodes and E is the set of semantic relations between them. Figure 5.1-A shows an example of a sentence in English, with its translations into Chinese, German, Italian and Spanish which have the same meaning and therefore the same abstract representation (Figure 5.1-C). Following state-of-the-art models for English AMR parsing [Zhang et al., 2019a], we tackle cross-lingual AMR parsing as a two-stage approach, i.e., concept and relation identification, which we briefly overview here and later detail in Section 5.2.2. For concept identification, given the sequence $T_L = (t_1, t_2, \ldots, t_j)$, t_i being a word in language L ($i \in \{1, \ldots, j\}$, $L \in \{\text{EN, DE, ES, IT, ZH}\}$), we train a neural network to generate the list of nodes $V = (v_1, v_2, \ldots, v_n)$, $v_i \in \text{English words} \cup \text{PropBank framesets} \cup \text{AMR keywords}$. In Figure 5.1-B we show the list of concepts that represent the words in the sentences of Figure 5.1-A. The relation identification procedure, instead, is inspired by the arc-factored approaches employed in dependency parsing [Kiperwasser and Goldberg, 2016], i.e., searching for the maximum-scoring connected subgraph over the identified concepts in the previous step. Thus, given the list of predicted nodes $V = (v_1, v_2, \ldots, v_n)$ and a learned score for each candidate edge, we search for the highest-scoring spanning tree and then merge the duplicate nodes based on unique node indices (see Section 5.2.2) to restore the final AMR graph. Figure 5.1-C shows the AMR representing the shared semantics of the sentences in Figure 5.1-A.



The city of Tel Aviv is fewer than 650 miles from Iranian territory.

Die Stadt Tel Aviv ist weniger als 650 Meilen vom iranischen Territorium entfernt.

La ciudad de Tel Aviv está a menos de 1.046 km del territorio iraní.

La città di Tel Aviv dista meno di 650 miglia dal territorio iraniano.

以色列最大的都会区特拉维夫市距离伊朗领土不到650英里。

Figure 5.1. Cross-Lingual AMR Parsing: (A) Sentences written in different languages sharing the same meaning; (B) concepts representing the words in the sentences; (C) the final AMR graph.

5.2.2 XL-AMR Model

XL-AMR is composed of two modules which are learned jointly, i.e., concept identification, modeled as a seq-to-seq problem, and relation identification, based on a biaffine attention classifier [Dozat and Manning, 2017]. We use a seq-to-seq model to dispose of the need for an AMR alignment module, i.e., word-to-node alignments. Lyu and Titov [2018] argue that alignments are important for injecting a useful inductive bias for AMR parsing and maintain that alignment-based parsers might be better than seq-to-seq for AMR parsing, owing to the relatively small amount of data available for AMR. However, aligning words to AMR nodes in cross-lingual parsing is challenging. The widely used AMR aligners are usually based on heuristics [Flanigan et al., 2014], or on the fact that AMR and English are highly cognate [Pourdamghani et al., 2014]. Hence, these approaches would not scale at large and neither be valid for cross-lingual alignment. Moreover, projecting the alignments across languages through English has shown to be noisy and to affect the parsing performance [Damonte and Cohen, 2018].

Concept identification At training time we obtain the list of nodes by first converting the graph into a tree, duplicating the nodes occurring in multiple relations, and then using a pre-order traversal over the tree. To account for reentrancies we assign a unique index to each node during traversal, similarly to Zhang et al. [2019a]. Following the attention-based encoder-decoder architecture proposed by Bahdanau et al. [2015], our concept identification module consists of a bidirectional RNN encoder and a decoder that attends to the source sentence at each concept decoding step.

The *encoder* employs an L-layer bidirectional RNN [Schuster and Paliwal, 1997] with LSTM cells [Hochreiter and Schmidhuber, 1997], i.e., BiLSTM, which encodes the input token embeddings e_i into hidden states h_i. Each hidden state $h_i^l = [\overrightarrow{h_i^l}; \overleftarrow{h_i^l}]$, is a concatenation of the forward hidden state and the backward hidden state at timestep i. Similarly to Zhang et al. [2019a], the input token embedding e_i is a concatenation of contextualized embeddings, word embeddings, Part-of-Speech (PoS) embeddings, token anonymization indicator[1] and character-level embeddings. The subsequent BiLSTM layer, instead, takes the hidden states of the previous layer as input.

The *decoder* also consists of L recurrent neural network (unidirectional) layers with LSTM cells. The decoder embedding layer concatenates word embeddings, node index embeddings and character-level embeddings. The layer l of the decoder calculates

$$d_t^l = \text{decoder}_l(d_t^{l-1}, d_{t-1}^l)$$

[1] Tokens representing named entities are anonymized during preprocessing and restored in postprocessing (Section 5.2.4).

where d_t^{l-1} is the concept hidden state of the previous layer at timestep t while d_{t-1}^l that of previous timestep. d_0^l is initialized with the concatenation of the encoder's last hidden states $h^l = [\overrightarrow{h^l}; \overleftarrow{h^l}]$. We follow the *input feeding* approach of Luong et al. [2015], which concatenates the output of the decoder's embedding layer and an attentional vector computed at the previous timestep. We first compute the *source attention distribution* a_t using additive attention [Bahdanau et al., 2015] as follows:

$$e_{t,i} = v^\top \text{tanh}(W_h h_i^L + W_s d_t^L + b_s)$$
$$a_t = \text{softmax}(e_t)$$
$$c_t = \sum_i a_{t,i} h_i$$

where v, W_h, W_s and b_s are model parameters, and c_t is the source context vector. Then, we compute the attentional vector,

$$\tilde{d}_t = \text{tanh}(W_c[c_t; d_t^L] + b_c)$$

where W_c and b_c are model parameters. Zhang et al. [2019a] used the attentional vector to allow the decoder to copy nodes predicted in the previous steps (*target-copy*), rather than only generating a new node from the vocabulary. As they provide empirical evidence that this is crucial for handling reentrancies, we employ their *target-copy* approach and use the attentional vector $\tilde{d}_t$ to:

i) feed in a dense layer and softmax to produce a probability distribution over the vocabulary $P_{vocab} = \text{softmax}(W_{vocab}\tilde{d}_t + b_{vocab})$;

ii) to learn a *target attention distribution* $\hat{a}_t$ (similar to the *source attention distribution* above);

iii) to calculate p_{copy} and $p_{generate}$ probabilities that decide either to copy one of the previously predicted nodes by sampling a node from the *target attention distribution* $\hat{a}_t$, or to generate a new node from the output vocabulary.

Each newly generated node is assigned a unique index, or it is assigned the index of the node copied from the previously generated concepts. At prediction time, we employ a beam search to decode the list of nodes based on the probability distribution computed above.

Relation identification For this module, we follow Zhang et al. [2019a] and use a deep biaffine classifier inspired by Dozat and Manning [2017], which takes as input the decoder states and factorizes the edge prediction in two components predicting i) whether there is an edge between a pair of nodes, and ii) the edge label for each possible edge, respectively. We direct the reader to Zhang et al. [2019a] and Dozat and Manning [2017] for technical details on the biaffine attention classifier. At prediction time, to ensure the validity of the tree, given the list of predicted nodes and the score for candidate edges, we search for the highest-scoring spanning tree using the Chu-Liu-Edmonds algorithm. We then merge the duplicate nodes based on the node indices to restore the final AMR graph.
The model is trained to jointly minimize the loss of reference nodes and edges.

5.2.3 Silver Data Creation

In order to train the cross-lingual AMR parser and to evaluate the cross-lingual properties of AMR as an interlingua, we project existing AMR annotations for English sentences to target language sentences following two different approaches.

Parallel sentences - silver AMR graphs. We follow Damonte and Cohen [2018] and project AMR graphs from English sentences to target language sentences through a parallel corpus. Differently from Damonte and Cohen [2018], we do not need *word-to-word* and *word-to-node* aligners for training the concept identification module, since we rely on a seq-to-seq translation model. Indeed, we directly pair a sentence in the target language with the AMR graph corresponding to its English counterpart. In this case, while the sentences are parallel, the AMR graphs are of *silver* standard quality, i.e., the English sentences of the parallel corpus are parsed using an existing AMR parser. We refer to this method as PARSENTS-SILVERAMR.

Gold AMR graphs - silver translations. In addition to pivoting through parallel sentences, we investigate whether considering human-annotated AMR graphs could bring more benefits than system produced AMR graphs. To this end, we make use of the existing gold standard datasets for AMR parsing, i.e., English sentence-AMR graph pairs, and use machine translation systems to translate the training sentences into the target language. This choice is motivated by the existence of reliable machine translation systems for the languages of our interest. Moreover, we validate the silver translations through a back-translation step [Sennrich et al., 2016]. That is, firstly, we translate the sentences from English to the target language and, secondly, using the same neural translation model, we translate the target language translations back to English. Then, to filter out less accurate translations we apply a 1-NN strategy based on the cosine similarity between translations and source sentence semantic embeddings, similarly to Artetxe and Schwenk [2019a]. If the nearest neighbour of a translation corresponds to its source English sentence, we consider it a good translation, otherwise we discard it. We employ semantic similarity since we have a two-step automatic translation, due to which lexical differences are introduced into translations compared to the original sentence. Typical machine translation metrics, e.g., BLEU or METEOR, rely on lexical similarity, which could lead good translations being discarded. In fact, we do not need the translation to be *word-to-word* aligned, but rather to preserve the meaning of the sentence, thus considering valid also the cases when certain words are translated into synonyms or related words. We refer to this method as GOLDAMR-SILVERTRNS.

5.2.4 Pre- and Postprocessing

AMR parsers in the literature rely on several pre- and postprocessing rules. We extend these rules for the cross-lingual AMR parsing task based on several multilingual resources such as Wikipedia, BabelNet [Navigli and Ponzetto, 2010; Navigli et al., 2021] [2], DBpedia Spotlight API [Daiber et al., 2013] for wikification in all languages but Chinese, for which we use Babelfy [Moro et al., 2014] instead, Stanford CoreNLP [Manning et al., 2014] for English preprocessing pipeline, the Stanza Toolkit [Qi et al., 2020] for Chinese, German and Spanish sentences, and Tint[3] [Aprosio and Moretti, 2016] for Italian.

The preprocessing steps consist of: i) lemmatization, ii) PoS tagging, iii) NER, iv) recategorization of entities and senses, v) removal of wiki links and polarity attributes. The postprocessing steps consist of restoring i) anonymized subgraphs, ii) wikification, iii) senses, iv) polarity attributes.

[2] Version 4.0

[3] Stanza does not provide a NER model for Italian.

Preprocessing. As NLP pipelines (steps i-iii) we use Stanford CoreNLP [Manning et al., 2014] for English sentences, the Stanza Toolkit [Qi et al., 2020] for Chinese, German and Spanish sentences, and Tint [4] [Aprosio and Moretti, 2016] for Italian. Recategorization and anonymization of entities is often used in English AMR parsing to reduce data sparsity [Zhang et al., 2019a; Lyu and Titov, 2018; Peng et al., 2017; Konstas et al., 2017]. Here we follow Konstas et al. [2017]; Zhang et al. [2019a] and anonymize entity subgraphs, which are identified by an AMR entity type and the `:name` role. First, the entity subgraphs are mapped with the corresponding text span in the sentence and then the text span is replaced with the anonymized token, i.e., `ENTITY_TYPE_i`. To match the entities in the AMR graphs, which are tied to English, with the corresponding text span in non-English sentences, we first collect all the possible lexicalizations of the entity in the target language using BabelNet [Navigli and Ponzetto, 2010]. It is a multilingual semantic network which brings together different resources such as WordNet, Wikipedia, etc., each node of which clusters together the lexicalizations that express the same concept in different languages. Then we search for the possible text spans in the sentence written in the target language. At test time, we anonymize the text spans which have been identified during the training data preprocessing and which are tagged by the NER tagger as entities.

Postprocessing. The anonymized subgraphs are restored using the anonymized text spans created during preprocessing. Then wiki links are restored using the DBpedia Spotlight API [5] [Daiber et al., 2013], commonly used in English AMR parsing [van Noord and Bos, 2017; Zhang et al., 2019a; Ge et al., 2019]. It provides models for multiple languages, except Chinese, for which we use Babelfy [Moro et al., 2014] Entity Linker. Since the wiki links identified by DBpedia Spotlight API are language-specific to the text, we further use Wikipedia inter-language links to retrieve the corresponding wiki links for the English entities. We restore senses as the most frequent sense of the predicate in the training data (using -01 if unseen), similar to Lyu and Titov [2018]; Zhang et al. [2019a], and finally restore polarity attributes based on heuristic rules observed on the training data and linguistic rules specific to each language.

[4] Stanza does not provide a NER model for Italian.

[5] http://github.com/dbpedia-spotlight/spotlight-docker.

DATASET	LANGUAGE	TRAIN INSTANCES	DEV INSTANCES	SOURCE
Gold	EN	36521	1368	AMR 2.0
PARSENTS SILVERAMR	DE	20000	2000	Europarl
	EN	20000	2000	Europarl
	ES	20000	2000	Europarl
	IT	20000	2000	Europarl
GOLDAMR SILVERTRNS	DE	34415	1319	AMR 2.0
	ES	34552	1325	AMR 2.0
	IT	34521	1322	AMR 2.0
	ZH	32154	1276	AMR 2.0

Table 5.1. Dataset quality standard, instances per language, and the source corpus of the sentences.

5.3 Experiments

We now present a set of experiments for cross-lingual AMR parsing when using different training techniques and the silver data we created (see Section 5.2.3). We discuss the results of our multiple settings and compare with previous approaches performing cross-lingual AMR parsing.

5.3.1 Dataset Creation Details

In Section 5.2.3, we explained the two projection approaches for obtaining cross-lingual AMR data, i.e., PARSENTS-SILVERAMR and GOLDAMR-SILVERTRNS.

For the first approach, inspired by Damonte and Cohen [2018], and for comparison purposes, we choose Europarl as parallel corpus.[6] We predict the silver AMR using the model of Zhang et al. [2019a].

For the second approach, instead, i.e., GOLDAMR-SILVERTRNS, we choose AMR 2.0 as gold dataset and translate the sentences into Chinese, German, Italian and Spanish. For German, Italian and Spanish, for both translating and back-translating the sentences we use the machine translation models made available by Tiedemann and Thottingal [2020, OPUS-MT].[7] For Chinese, instead, since OPUS-MT does not provide translation models, we employ the released MASS [8] [Song et al., 2019b] supervised neural translation models. Then, to filter out less accurate translations, we compute the cosine similarity between dense

[6] We do not produce silver AMR graphs for Chinese since Europarl does not cover the Chinese language.

[7] We provide the list of models we used in Appendix B.1.

[8] http://github.com/microsoft/MASS/tree/master/MASS-supNMT

semantic representations of the original English sentence and its back-translated counterpart. To embed the sentences we use LASER [Artetxe and Schwenk, 2019b], a state-of-the-art model for sentence embeddings. Details on the number of instances per language and for each silver data approach are shown in Table 5.1.

5.3.2 Evaluation Benchmark

We evaluate on the *Abstract Meaning Representation 2.0 - Four Translations* [Damonte and Cohen, 2020], a corpus containing translations of the test split of 1371 sentences from the LDC2017T10 (AMR 2.0), in Chinese (ZH), German (DE), Italian (IT) and Spanish (ES).

5.3.3 Evaluation Metrics

Evaluating the performance of the parser requires comparing two unaligned graphs. Similar to Chapter 4, we evaluate the systems according to the overall Smatch score and also their performance in separate phenomena. Briefly summarizing, Smatch [Cai and Knight, 2013] computes the degree of overlap of two AMR graphs in terms of logical triples overlap. [9] Since the two graphs are not aligned, an integer programming technique is required to approximate the best match, which best aligns the variables whose names could differ between two AMR graphs. To compare multiple predicated AMRs and the gold AMRs, the macro-averaged F1 is used.

In addition to the aggregated Smatch metric, we evaluate the parsers using the AMR-evaluation tools [10] developed by Damonte et al. [2017] through which we can perform a fine-grained analysis based on several separate phenomena, e.g., SRL, reentrancy.

5.3.4 Training and Tuning

We train XL-AMR following different strategies:

- **Zero-shot** – the model is trained on English sentences only, relying on multilingual features, and is evaluated on all the target languages (henceforth ∅-shot).
- **Language-specific** – the model is trained only on target language data, i.e., DE, ES, IT or ZH, and evaluated in the same language.

[9] http://github.com/snowblink14/smatch

[10] http://github.com/mdtux89/amr-evaluation

- **Bilingual** – the model is trained on English data and one of either DE, ES, IT or ZH, and evaluated in the target language.
- **Multilingual** – the model is trained on data from all available languages per setting and evaluated on the target languages.

We denote these variations of XL-AMR, as XL-AMR^{data} where, $data \in \{par, trans, amr\}$, par referring to the data produced with PARSENTS-SILVERAMR approach, $trans$ to GOLDAMR-SILVERTRNS approach, amr to the AMR 2.0 English gold standard, and $data$+ refers to combining par or $trans$ with amr. We provide details of our model hyperparameters in Appendix B.2.

5.3.5 Comparison Systems

We first compare all the XL-AMR system variations among them, identifying their advantages and disadvantages. Then, we compare with Damonte and Cohen [2018, AMREAGER Multilingual] (henceforth AMREAGER), the only existing cross-lingual AMR parser at the time of writing. In particular, we compare the results of the XL-AMR variants with the projection method of AMREAGER on the gold dataset, i.e., *AMR 2.0 - Four Translations*. We remark that we do not consider the results of their Machine Translation[11] method, since, as emphasised by the authors, it is not informative in terms of cross-lingual properties of AMR [Damonte and Cohen, 2018] because it performs English AMR parsing.

5.4 Results

In Table 5.2 we show performance of the models in terms of Smatch. We point out the low score of the $\emptyset$-shot models, i.e., XL-AMR$_{\emptyset}^{amr}$ and XL-AMR$_{\emptyset}^{par+}$, which perform lower than AMREAGER, especially in the Chinese language. However, XL-AMR$_{\emptyset}^{par+}$ noticeably improves over XL-AMR$_{\emptyset}^{amr}$, which can be explained by the fact that seq-to-seq requires a large amount of data in order to generalize. This is confirmed by a fine-grained analysis showing lower accuracy of XL-AMR$_{\emptyset}^{amr}$ compared to XL-AMR$_{\emptyset}^{par+}$ in concept identification, which, we recall, is a seq-to-seq module.

[11] It translates the test sentences from the target language to English and parses the translations using an English parser.

PARSER	CONFIGURATION	DE	ES	IT	ZH
AMREAGER	Language-Specific	39.0	42.0	43.0	35.0
XL-AMR$_{\emptyset}^{amr}$	$\emptyset$-shot	32.7	39.1	37.1	25.9
XL-AMR$_{\emptyset}^{par+}$	$\emptyset$-shot	38.3	41.8	41.0	23.9
XL-AMR^{par}	Language-Specific	40.8	44.2	43.4	-
	Multilingual	41.5	45.6	45.0	-
	Bilingual	42.7	47.9	46.7	-
XL-AMR^{par+}	Multilingual	46.3	51.2	50.9	-
	Bilingual	47.0	53.0	51.4	-
XL-AMR^{trans}	Language-Specific	51.6	56.1	56.7	**43.1**
	Multilingual	49.9	53.0	54.0	40.0
	Multilingual (-ZH)	51.5	55.5	55.9	-
XL-AMR^{trans+}	Multilingual	49.9	53.2	53.5	41.0
	Multilingual (-ZH)	52.1	56.2	56.7	-
	Bilingual	**53.0**	**58.0**	**58.1**	41.5

Table 5.2. Smatch F1 scores on DE, ES, IT and ZH. Best scores per language are denoted in **bold**.

Interestingly, the language-specific XL-AMR^{par}, even if trained on less instances, outperforms the $\emptyset$-shot models by a large margin. Moreover, it also surpasses AMREAGER, which is trained on the same sentences from Europarl. The results are further improved when jointly training in multiple languages, i.e., when using the multilingual and bilingual configurations. We attribute this improvement to the ability of a seq-to-seq model to learn better when provided with a larger training set. The domain of the Europarl data is very specific, which does not enable the model to generalize in sentences from other domains. In fact, the XL-AMR^{par+} models significantly improve over the XL-AMR^{par} bilingual and multilingual models. We attribute the higher performances of XL-AMR^{par+} to i) larger training dataset, ii) training on different domains, and iii) better quality of the data (AMR 2.0 data is human annotated).

The XL-AMR^{trans} models perform best: we note that the performances of the language-specific variants outperform those of the multilingual XL-AMR^{trans} models, in contrast to the behaviour of the XL-AMR^{par} models, suggesting that the addition of silver data in other languages is not beneficial. This may be due to the fact that the AMR graphs of translated sentences are the same, thus as a consequence the model does not access extra information. Moreover, the inclusion of translated sentences in other languages slightly

	AMREAGER				XL-AMR^{par+}				XL-AMR^{trans+}			
Metric	DE	ES	IT	ZH	DE	ES	IT	ZH	DE	ES	IT	ZH
SMATCH	39.1	42.1	43.2	34.6	47.0	53.0	51.4	-	**53.0**	**58.0**	**58.1**	**43.1**
Unlabeled	45.0	46.6	48.5	41.1	52.0	58.3	57.1	-	**57.7**	**63.0**	**63.4**	**48.9**
No WSD	39.2	42.2	42.5	34.7	47.1	53.2	51.5	-	**53.2**	**58.4**	**58.4**	**43.2**
Reentrancies	18.6	27.2	25.7	15.9	33.6	40.1	39.2	-	**39.9**	**46.6**	**46.1**	**34.7**
Concepts	44.9	53.3	52.3	39.9	48.7	58.0	55.6	-	**58.0**	**65.9**	**64.7**	**48.0**
Named Ent.	63.1	65.7	67.7	67.9	63.1	61.6	62.7	-	**66.0**	**66.2**	**70.0**	60.6
Wikification	49.9	44.5	50.6	46.8	61.4	63.8	66.1	-	**60.9**	**63.1**	**67.0**	**54.5**
Negation	**18.6**	19.8	22.3	6.8	8.1	21.5	25.7	-	11.7	**23.4**	**29.2**	**12.8**
SRL	29.4	35.9	34.3	27.2	40.8	48.7	46.7	-	**47.9**	**55.2**	**54.7**	**41.3**

Table 5.3. Fine-grained F1 scores DE, ES, IT and ZH. Best scores per language are denoted in **bold**.

harms the performances. This is confirmed by the removal from the training set of the most distant language, in the multilingual (-ZH) model, which in turn achieves around 2 F1 points more compared to the multilingual version including Chinese. This can be further explained by the linguistic differences between Chinese and the other languages, which prevent them from benefiting from the inclusion of Chinese instances in the training set. However, when adding English gold AMR 2.0, i.e., XL-AMR^{trans+}, the model benefits from the better quality of this dataset. In fact, the bilingual version of XL-AMR^{trans+} is the best performing across the board in German, Spanish and Italian, surpassing AMREAGER by at least 14 F1 points and both XL-AMR^{par} and XL-AMR^{par+} by at least 5 F1 points in each language. Interestingly, the best results in Chinese are achieved by the language-specific XL-AMR^{trans} surpassing AMREAGER by 8 F1 points and the $\emptyset$-shot models by more than 17 F1 points. This is once again explained by the linguistic differences of Chinese as compared to the other languages, which render the additional data non-beneficial.

5.4.1 Fine-grained Results

Table 5.3 shows the fine-grained evaluation of AMREAGER and our best performing models for each data creation approach, for which we use the evaluation tools of Damonte et al. [2017]. The fine-grained results for the AMREAGER are not reported by Damonte and Cohen [2018], therefore we run the evaluation using their released models.[12] Our best model outperforms AMREAGER in all sub-tasks except for *Negations* in German and *Named Entities*

[12] http://github.com/mdtux89/amr-eager-multilingual

in Chinese, which are prone to heuristic string matching errors in the pre- and postprocessing procedure of our models. XL-AMR^{trans+} achieves significantly higher performance in *Reentrancies*, *Concepts*, *SRL*, in all the tested languages, compared to AMREAGER, thus demonstrating the effectiveness of our parser and data creation approaches.

In summary, translating the gold standard training data, i.e., GOLDAMR-SILVERTRNS, leads XL-AMR to achieve higher performances than when trained on parallel sentences associated with silver AMR graphs, i.e., PARSENTS-SILVERAMR.

5.5 Analysis

We manually check the predictions of XL-AMR in order to establish the nature of the mistakes based on the Smatch score between the gold and predicted AMR graphs and determine their severity. Then, we observe how XL-AMR handles the translation divergences, i.e., linguistic distinctions that make transfer across languages difficult [Dorr, 1994].

5.5.1 Smatch Errors

The parser has difficulties with some compounded words in German, e.g., *Uranproduktionsfähigkeit* (*uranium production capability*), *Kernkraftstoffkreislauf* (*nuclear fuel cycle*), for which it fails to break their meaning down to the correct subgraph (a), but instead predicts a generic node (b).

(a)

```
(c / cycle-02
    :ARG1 (f /fuel
        :mod (n / nucleus)))
```

(b)

```
(t / thing)
```

This issue can be alleviated using a better preprocessing to split the compounds.

Several cases with low Smatch score are due to inconsistent translations of test set sentences into the target language, even though, we recall, the test set has been manually translated. This could be due to translator choices, but can lead to divergent meaning structures, e.g., *Ich kann verstehen, wie Du Dich fühlst (DE)* (*I can understand how you are feeling*) whose original English sentence from which the AMR graph is projected is *I know what you're* *feeling*. The gold AMR graph is thus not appropriate for the German sentence, due to the sentence's different meaning. Thus these mistakes are not due to the parser, but to the translations. An interesting cause of drop in the Smatch arises from the prediction of concepts that are synonyms of the corresponding concepts in the gold graph, e.g., say-01 → state-01, stop-01 → halt-01, best friend → best mate, demand-01 → urge-01, etc. We notice that the predicted concepts (to the left of the arrow) are less specific than the gold concepts, yet somehow preserve the meaning. These examples show that the parser captures a close meaning even when failing to predict the exact concept.

5.5.2 Translation Divergences

We investigate how XL-AMR deals with the cases where there exist translation divergences, i.e., cases in which source and target language have different syntactic ordering properties [Dorr, 1990], as classified by Dorr [1994] using the following 7 categories: i) *thematic*, ii) *promotional*, iii) *demotional*, iv) *structural*, v) *conflational*, vi) *categorial*, vii) *lexical*.[13]

Thematic divergence. A *thematic* divergence happens when the argument-predicate structure is different across languages, e.g., ***I** like travelling* where *I* is the subject, in Italian becomes ***Mi** piace viaggiare*, and *Mi* is now the object. XL-AMR overcomes this divergence and predicts the correct AMR:

```
(l / like-01
      :ARG0 (i / I)
      :ARG1 (t / travel
                  :ARG0 i))
```

Promotional & demotional. These two divergences can be merged into the *head switching* macro-category. They arise when a modifier in one language is promoted to a main verb in the other, or vice versa, e.g., *John **usually** goes home* is *Juan **suele** ir a casa* (*John is accustomed to go home*) in Spanish. XL-AMR correctly parses the sentence as:

[13] In absence of a larger available resource for language divergences, here we make use of some of the pre-classified examples from Dorr [1990, 1994].

```
(g / go-01
    :ARG0 (p / person
        :name (n / Juan))
    :ARG4 (h / home)
    :mod (u / usual))
```

Structural. A *structural* divergence exists when a verbal object is realized as a noun phrase (NP) in one language and as prepositional phrase (PP) in the other, e.g., *I saw **John*** where *John* is NP, is translated as *Vi **a Juan*** (*I saw to John*) in Spanish where *a Juan* is PP. This also is not a problem for our parser, which predicts the correct graph:

```
(s / see-01
    :ARG0 (i / I)
    :ARG1 (p / person
        :name (n / Juan)))
```

Conflational. A *conflational* divergence refers to the translation of two or more words in one language into one word in the other. The above errors in German compounded words fall into this category and our model does not handle them properly. However, regarding other languages this problem is not common, e.g., *I **fear*** translates into *Io* ***ho paura*** (*I have fear*) in Italian and the parser correctly predicts the AMR graph:

```
(f / fear-01
    :ARG0 (i / I))
```

Categorical. A *categorical* divergence arises when the same meaning is expressed by different syntactic categories across languages, e.g., *I **agree***, where *agree* is a *verb*, is expressed by a *noun* in Italian and Spanish, *Sono d'**accordo*** and *Estoy de **acuerdo***. The parser correctly predicts the same AMR for both languages:

```
(a / agree-01
    :ARG0 (i / I))
```

Lexical A *lexical* divergence arises when a verb in the source language is translated with a different lexical verb, e.g., *Juan **broke** into the room*, *Juan **forzó** la entrada al cuarto*, in which the verb *break* in English is translated with the verb *forzar* (*force*) in Spanish. XL-AMR predicts the following graphs for the English (a) and Spanish (b) sentences:

(a)

```
(f / break-02
      :ARG0 (p / person
            :name (n / Juan))
      :ARG1 (r / room)))
```

(b)

```
(f / force-01
      :ARG0 (p / person
                  :name (n / Juan))
      :ARG2 (e / enter-01
                  :ARG0 p
                  :ARG1 (r / room)))
```

This, even though it is correctly parsed, does not overcome the lexical difference of the action, which results in different AMR graphs for the same meaning. This is partially due to the fact that AMR is bound to lexical forms in English.

In summary, XL-AMR overcomes most of the foregoing structural divergences with the exception of two cases: i) the conflational divergence in German, that is caused by the language's compound words vocabulary, for the resolution of which a better preprocessing can be beneficial; ii) the lexical divergence that persists despite the parser predicting a valid graph. The latter divergence results in non-parallel structures for parallel meanings, and we believe this might be tackled by integrating a unified ontology for synonyms or related meanings within the AMR formalism, along the line of disjunctive AMR[14] [Banarescu et al., 2013]. We leave exploration of this approach open for future work. Furthermore, we notice that recent work on translation divergences could be considered for a deeper analysis such

[14] http://amr.isi.edu/damr.1.0.pdf

as Deng and Xue [2017]; Vyas et al. [2018]; Nikolaev et al. [2020]; Briakou and Carpuat [2021], and in the context of AMR parsing, Wein and Schneider [2021].

5.6 Summary

In this Chapter, we explored transfer learning techniques to enable high-performance cross-lingual AMR parsing. We created silver data based on annotation projection through parallel sentences and machine translation, on which we trained XL-AMR, a cross-lingual AMR parser that achieves the highest results reported to date on Chinese, German, Italian and Spanish. A qualitative evaluation showed that XL-AMR can handle most of the structural divergences among languages. The performance of XL-AMR together with the qualitative analysis suggests that carefully modeling cross-lingual AMR parsing leads to the production of suitable AMR structures across languages. XL-AMR and the advancements in the field following to it [Procopio et al., 2021; Sheth et al., 2021; Uhrig et al., 2021; Cai et al., 2021b,a], encourage us to extend this line of our research, also by exploiting more considerable multilingual semantic resources, to further improve the parsing quality. Moreover, the improvements in performance for cross-lingual AMR parsing open the other research direction of integrating AMR into downstream cross-lingual applications to investigate their added value.

∼ This page was intentionally left blank ∼

Chapter 6

Conclusions

6.1 Overview

In this final Chapter, we briefly summarize the main topics and contributions of this dissertation and then discuss the open directions and future perspectives emerging from the findings of our thesis, and not only.

We first described the similarities and differences of the SRL and AMR, distinguishing their advantages and disadvantages in representing sentence semantics (Chapter 1). Moreover, we overviewed and discussed the complexity of these formalisms and the tendency of the existing approaches for proposing complex task-specific architectures that rely on long pipelines and intrinsic data-specific heuristics for achieving high performances (Chapter 2). Then, we identified the gaps and open questions in dependency- and span-based SRL, such as the unexplored potential of seq-to-seq approaches and the recent convergence of existing models in the same performance pool, and reformulated SRL as a sequence generation task to jointly generate predicate senses and semantic roles. We achieved significantly higher performances when compared to the existing non-end-to-end seq-to-seq approaches, thus reaching state-of-the-art results previously obtained through task-specific sequence labeling approaches (Chapter 3).

Furthermore, we recognized the weaknesses of previous seq-to-seq approaches to AMR parsing, and by devising novel compact graph linearizations and exploiting pretrained encoder-decoder models, we achieved state-of-the-art results both in AMR parsing and generation. Indeed, we cast these tasks as symmetric tasks, similar to the case for machine translation from one language to another. We further made our state-of-the-art systems available to the community through a highly interactive, easy-to-use Web Interface and RESTful APIs equipped with a feedback mechanism that aspires to enable active learning in AMR (Chapter 4).

Finally, we addressed the questions on the suitability of AMR to represent meaning across languages by using English-centric structures as interlingua in Italian, Spanish, German, and Chinese. Indeed, we explored different transfer learning techniques to overcome the challenges posed by the paucity of data for cross-lingual AMR parsing, and relied on seq-to-seq models for translating from non-English sentences, to English-centric AMR concepts, thus disposing of noisy word-to-node AMR alignments (Chapter 5).

6.2 Future Work and Perspectives

6.2.1 Mid-term Perspective

Some mid-term directions that are worth mentioning and that are inspired by the work presented in this thesis are the following:

Ensembling of fundamentally different SRL approaches. Recent techniques, despite their differences, seem to have plateaued in terms of performance. However, through a simple analysis of two radically different SRL approaches, i.e., sequence labeling and seq-to-seq learning, we showed that, despite their similar results and learning curves, their behavior is different, e.g., in precision and recall. One popular yet often overlooked approach to improve the results of an approach is ensembling. While there have been a handful of attempts using ensembling techniques for SRL [Ouchi et al., 2018b], they benefit from combining the predictions coming from different initializations of the same model. We argue that the ensembling of radically different approaches could be more effective than an ensemble of models employing the same learning paradigm. To this end, one potential direction is devising a strategy to guide the respective model to learn from their mistakes, provided that these approaches are probably complementary to each other. A relevant work for ensembling different approaches to graph predictions with AMR was recently presented by Lam et al. [2021].

Joint learning of SRL and AMR. Throughout this dissertation, we describe the overlapping aspects between SRL and AMR. Indeed, both annotate the predicate-argument structure relying on the same predicate inventory and, therefore, the same linguistic theory. Moreover, AMR is composed of multiple subtasks of NLP such as WSD, NER and EL, which cover a wide range of successful approaches in the literature [Bevilacqua et al., 2020; Barba et al., 2021; Wu et al., 2020; Cao et al., 2021]. In our works, we reformulate the graph-like structures of SRL and AMR as sequences, which allows us to employ similar architectures for learning both. Similarly, Bevilacqua et al. [2020] and Cao et al. [2021] propose sequence generation approaches for WSD and EL, respectively. To this end, one potential direction is exploring multitask learning techniques for jointly learning AMR graphs and the subtasks it covers. Joint modeling of semantic parsers has been previously tackled by Peng et al. [2018], who combine frame-semantic parsing and semantic dependency parsing, achieving improvements in both. Previous multitasking approaches in the context of AMR instead, mainly include non-semantic tasks such as syntactic parsing and machine translation [Xu et al., 2020]. We argue that semantic tasks such as SRL, WSD, NER and EL, which moreover are subtasks of the AMR formalism, could be more beneficial for AMR parsing and generation. Indeed, while we enable AMR to benefit from the specialization of its subtasks, we also allow the latter to take advantage of the richer semantics included in AMR.

Extending the applicability of AMR as an interlingua. In this thesis, we analyzed the suitability of English-centric AMR to represent the meaning of sentences across languages. Our findings through this work immediately gained the attention of researchers, and XL-AMR was followed by several successful research works which significantly raise the performance in cross-lingual AMR parsing in the benchmark comprising of Chinese, German, Italian, and Spanish translations [Procopio et al., 2021; Sheth et al., 2021; Cai et al., 2021b; Uhrig et al., 2021; Xu et al., 2021]. However, it is not clear to what extent AMR can be used as an interlingua. For instance, the performance trend of cross-lingual parsers in Italian and Spanish is higher than that achieved in German and especially Chinese. This result is indeed expected because Chinese is the most distant language concerning the others in this benchmark. Similarly, the structure of the sentences in German is different from what we see in English or Italian, e.g., the verb is placed at the end of the sentence. To this end, one potential direction would be to analyze the suitability of AMR to act as an interlingua in inter-language groups. Fan and Gardent [2020] conduct probably the most interesting study to this direction, who investigate how the AMR generation results are affected by the set of languages used for training the multilingual models.

Semantically-enhanced applications. In Section 2.6 we outlined the applications in which AMR structures have been integrated with encouraging results. While most of the applications have relied on older parsers, the performance of the recent AMR parsers has increased by around 10 Smatch F1 points over the past three years. Moreover, in our SPRING work, we assess the generalization abilities of our systems in an out-of-distribution setting, which better mimics the open-world data. These improvements feed the hope that AMR systems are mature to produce good enough structures that can be more beneficial in downstream applications such as Machine Translation and Question Answering.

6.2.2 Long-term Perspective

Some long-term directions that are worth mentioning and that are inspired by the work presented in this thesis are the following:

Improving by active learning. In SPRING Online Services, we introduce a simple feedback mechanism that allows users to submit their modification to the system's outputs (see Section 4.6.1). We believe that collecting user validation of AMR graphs can be critical to future developments of parsers via active leaning [Settles, 2009]. Indeed, by analyzing the nature of modifications, researchers might develop new strategies for handling edge cases or the generalizability of the models in real-world data.

Devising a truly semantic interlingua. In this thesis, we assumed that AMR can be used as an interlingua by projecting English-centric AMR graphs to parallel sentences in multiple languages. However, meaning representations should not revolve mainly around English. Indeed, there are several challenges that prevent AMR from being a *truly* semantic formalism, making it inadequate to act as an interlingua, such as i) the intersection of AMR concepts' vocabulary and the English lexicon (e.g., *doctor, medicine*), and ii) the extensive use of the PropBank verbal framesets (e.g., *tell.01, take.04*), from which it also takes the core predicate argument roles (e.g., `:ARG1`, `:ARG2`). The latter, we recall from Section 2.3, is available in a limited number of languages, and even when similar predicate inventories do exist, they rely on language-specific rules and theories. Moreover, the usage of a lexicon makes AMR not fully semantic since words are not only ambiguous but also language-specific. We believe that to get closer to solving the puzzle of Natural Language Understanding, we need a *truly* semantic language-independent representation, with concepts drawn from a multilingual inventory, e.g., BabelNet, and semantic relations being shared across languages. Indeed, an interlingua would be helpful not only at a practical modeling level, since it requires a unified representation for text in all languages instead of multiple language-specific ones, but also at the broader application level, e.g., interlingual Machine Translation [Richens, 1958]. From another point of view, the idea of an interlingua could lead to philosophical discussions or face several practical issues. Indeed, having an interlingua assumes a universal organization of meanings, which might not hold from one language to another. At the same time, it could be challenging to define what is a lexical item in a concept ontology, how their senses are delineated, and what action to take when a concept does not have a direct lexicalization across languages. Nonetheless, making attempts towards the idea of an interlingua by using the available multilingual resources could, at the very least, reduce the English-specific bias in representing meaning.

Interpreting through conceptual representations. As AI-enabled systems have become ever more accurate and advanced, it is difficult for humans to comprehend the calculation process that led to certain decisions. Indeed, those models created directly from data are commonly referred to as “black box" and are hard to interpret. On the other hand, people are embracing AI-powered systems in different areas of their life. For this reason, understanding how these systems work – or if they are working as expected – brings essential advantages. We believe that conceptual meaning representations can facilitate the path towards machine interpretability. One potential direction could be to use meaning representations as an intermediate layer, which decouples text comprehension from text generation, using meaning to text generation systems (similar to SPRING). Doing so might increase systems interpretability because meaning representations are both human-readable and machine-processable.

Bibliography

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Izaskun Aldezabal, María Jesús Aranzabe, Arantza Díaz de Ilarraza, and Ainara Estarrona. 2010. Building the Basque PropBank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Alessio Palmero Aprosio and Giovanni Moretti. 2016. Italy goes to stanford: a collection of corenlp modules for italian. *CoRR*, abs/1609.06204.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Yoav Artzi, Nicholas Fitzgerald, and Luke Zettlemoyer. 2014. Semantic parsing with Combinatory Categorial Grammars. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, Doha, Qatar. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua 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*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, 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, Sofia, Bulgaria. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceed-*

ings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Beck, Gholamreza Haffari, and Trevor 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, Melbourne, Australia. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021a. One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021b. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for Hindi/Urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 186–189, Suntec, Singapore. Association for Computational Linguistics.

Rexhina Blloshmi, Michele Bevilacqua, Edoardo Fabiano, Valentina Caruso, and Roberto Navigli. 2021a. SPRING Goes Online: End-to-End AMR Parsing and Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana. Association for Computational Linguistics.

Rexhina Blloshmi, Simone Conia, Rocco Tripodi, and Roberto Navigli. 2021b. Generating senses and roles: An end-to-end model for dependency- and span-based semantic role labeling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3786–3793. International Joint Conferences on Artificial Intelligence Organization.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2487–2500, Online. Association for Computational Linguistics.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020a. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Claire Bonial, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare Voss. 2020b. InfoForager: Leveraging semantic search with AMR for COVID-19 research. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 67–77, Barcelona Spain (online). Association for Computational Linguistics.

Eleftheria Briakou and Marine Carpuat. 2021. Beyond noise: Mitigating the impact of fine-grained semantic divergences on neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7236–7249, Online. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2019. Core semantic first: A top-down approach for AMR parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3799–3809, Hong Kong, China. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020a. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020b. Graph transformer for graph-to-sequence learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7464–7471.

Deng Cai, Xin Li, Jackie Chun-Sing Ho, Lidong Bing, and Wai Lam. 2021a. Multilingual AMR parsing with noisy knowledge distillation. *CoRR*, abs/2109.15196.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rui Cai and Mirella Lapata. 2019a. Semi-supervised Semantic Role Labeling with cross-view training. In *Proc. of EMNLP*.

Rui Cai and Mirella Lapata. 2019b. Syntax-aware Semantic Role Labeling without parsing. *Transactions of ACL*.

Shu Cai 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, Sofia, Bulgaria. Association for Computational Linguistics.

Yitao Cai, Zhe Lin, and Xiaojun Wan. 2021b. Making better use of bilingual information for cross-lingual AMR parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1537–1547, Online. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic Role Labeling. In *Proc. of CoNLL*.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine Learning*.

Xinchi Chen, Chunchuan Lyu, and Ivan Titov. 2019. Capturing argument interaction in Semantic Role Labeling with capsule networks. In *Proc. of EMNLP*.

Simone Conia, Andrea Bacciu, and Roberto Navigli. 2021. Unifying cross-lingual semantic role labeling with heterogeneous linguistic resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–351, Online. Association for Computational Linguistics.

Simone Conia and Roberto Navigli. 2020. Bridging the gap in multilingual semantic role labeling: a language-agnostic approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1396–1410, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *I-SEMANTICS 2013 - 9th International Conference on Semantic Systems, ISEM '13, Graz, Austria, September 4-6, 2013*, pages 121–124.

Marco Damonte and Shay Cohen. 2020. *Abstract Meaning Representation 2.0 - Four Translations LDC2020T07*. Web Download, Philadelphia: Linguistic Data Consortium.

Marco Damonte and Shay B. Cohen. 2018. 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, Volume 1 (Long Papers)*, pages 1146–1155, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for AMR-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3649–3658, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Damonte, 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, Valencia, Spain. Association for Computational Linguistics.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Donald Davidson. 1969. *The Individuation of Events*, pages 216–234. Springer Netherlands, Dordrecht.

Angel Daza and Anette Frank. 2018. A sequence-to-sequence model for semantic role labeling. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 207–216, Melbourne, Australia. Association for Computational Linguistics.

Angel Daza and Anette Frank. 2019. Translate and label! an encoder-decoder approach for cross-lingual semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 603–615, Hong Kong, China. Association for Computational Linguistics.

Dun Deng and Nianwen Xue. 2017. Translation divergences in Chinese–English machine translation: An empirical investigation. *Computational Linguistics*, 43(3):521–565.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bonnie Dorr. 1990. Solving thematic divergences in machine translation. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 127–134, Pittsburgh, Pennsylvania, USA. Association for Computational Linguistics.

Bonnie J. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Magali Sanches Duran and Sandra Maria Aluísio. 2011. Propbank-br: a Brazilian Portuguese corpus annotated with semantic role labels. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.

Angela Fan and Claire 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. Association for Computational Linguistics.

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1001–1007, Online. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime 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 (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

DongLai Ge, Junhui Li, Muhua Zhu, and Shoushan Li. 2019. Modeling source syntax and semantics for neural AMR parsing. In *Proc. of IJCAI*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*.

Michael Wayne Goodman. 2020. 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*, pages 312–319, Online. Association for Computational Linguistics.

Saurabh Gupta and Jitendra Malik. 2015. Visual Semantic Role Labeling. *arXiv preprint*.

Jan Hajič, Ondřej Bojar, and Zdeňka Urešová. 2014. Comparing Czech and English AMRs. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, pages 55–64, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

Hardy Hardy and Andreas Vlachos. 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*, pages 768–773, Brussels, Belgium. Association for Computational Linguistics.

Katri Haverinen, Jenna Kanerva, Samuel Kohonen, Anna Missilä, Stina Ojala, Timo Viljanen, Veronika Laippala, and Filip Ginter. 2015. The finnish proposition bank. *Language Resources and Evaluation*, 49(4):907–926.

Shexia He, Zuchao Li, and Hai Zhao. 2019. Syntax-aware multilingual semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5350–5359, Hong Kong, China. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Fuad Issa, Marco Damonte, Shay B. Cohen, Xiaohui Yan, and Yi Chang. 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)*, pages 442–452, New Orleans, Louisiana. Association for Computational Linguistics.

Aishwarya Kamath and Rajarshi Das. 2019. A survey on semantic parsing. In *Automated Knowledge Base Construction (AKBC)*.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir R. Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. In *Proc. of NAACL*.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Rohit J. Kate and Yuk Wah Wong. 2010. Semantic parsing: The task, the state of the art and the future. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 6, Uppsala, Sweden. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Hideo Kobayashi and Vincent Ng. 2020. Bridging resolution: A survey of the state of the art. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3708–3721, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke 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. Association for Computational Linguistics.

Hoang Thanh Lam, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam M. Nguyen, Dzung T. Phan, Vanessa López, and Ramon Fernandez Astudillo. 2021. Ensembling graph predictions for amr parsing. In *Advances in Neural Information Processing Systems 35: Annual*

Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bin Li, Yuan Wen, Weiguang Qu, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 7–15, Berlin, Germany. Association for Computational Linguistics.

Tao Li, Parth Anand Jawale, Martha Palmer, and Vivek Srikumar. 2020. Structured tuning for semantic role labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8402–8412, Online. Association for Computational Linguistics.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform Semantic Role Labeling. In *Proc. of AAAI*.

Kexin Liao, Logan Lebanoff, and Fei 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. Association for Computational Linguistics.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. 2020. I know what you asked: Graph path learning using AMR for commonsense reasoning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2459–2471, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the*

Association for Computational Linguistics, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020a. On the variance of the adaptive learning rate and beyond. In *Proc. of ICLR*.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020b. On the variance of the adaptive learning rate and beyond. In *ICLR 2020*.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An AMR aligner tuned by transition-based parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2430, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher 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. Association for Computational Linguistics.

Chunchuan Lyu, Shay B. Cohen, and Ivan Titov. 2019. Semantic role labeling with iterative structure refinement. In *Proc. of EMNLP*.

Chunchuan Lyu, Shay B. Cohen, and Ivan Titov. 2020. A differentiable relaxation of graph segmentation and alignment for AMR parsing. *Arxiv preprint*, abs/2010.12676.

Chunchuan Lyu 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)*, pages 397–407, Melbourne, Australia. Association for Computational Linguistics.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim 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. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. 2018a. Exploiting semantics in neural Machine Translation with graph convolutional networks. In *Proc. of NAACL*.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. 2018b. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware Semantic Role Labeling. In *Proc. of EMNLP*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Semantic Role Labeling: An introduction to the special issue. *Computational Linguistics*.

Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Special issue introduction: Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159.

Jonathan May and Jay Priyadarshi. 2017. SemEval-2017 task 9: Abstract Meaning Representation parsing and generation. In *Proceedings of the 11th International Workshop on*

Semantic Evaluation (SemEval-2017), pages 536–545, Vancouver, Canada. Association for Computational Linguistics.

Noelia Migueles-Abraira, Rodrigo Agerri, and Arantza Diaz de Ilarraza. 2018. Annotating Abstract Meaning Representations for Spanish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Tahira Naseem, Abhishek Shah, Hui Wan, Radu Florian, Salim Roukos, and Miguel Ballesteros. 2019. Rewarding Smatch: Transition-based AMR parsing with reinforcement learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4586–4592, Florence, Italy. Association for Computational Linguistics.

Roberto Navigli. 2018. Natural Language Understanding: Instructions for (present and future) use. In *Proc. of IJCAI*.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of babelnet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4559–4567. ijcai.org.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Dmitry Nikolaev, Ofir Arviv, Taelin Karidi, Neta Kenneth, Veronika Mitnik, Lilja Maria Saeboe, and Omri Abend. 2020. Fine-grained analysis of cross-linguistic syntactic divergences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1159–1176, Online. Association for Computational Linguistics.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018a. A span selection model for Semantic Role Labeling. In *Proc. of EMNLP*.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018b. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Martha Palmer, Olga Babko-Malaya, Ann Bies, Mona Diab, Mohamed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A pilot Arabic Propbank. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 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, Denver, Colorado. Association for Computational Linguistics.

Kishore Papineni, 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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1492–1502, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. 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*, pages 366–375, Valencia, Spain. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

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

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Vivek Srikumar. 2021. Supertagging the long tail with tree-structured decoding of complex categories. *Transactions of the Association for Computational Linguistics*, 9:243–260.

Luigi Procopio, Rocco Tripodi, and Roberto Navigli. 2021. SGL: Speaking the graph languages of semantic parsing via multilingual translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 325–337, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. 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, pages 101–108, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. -.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Arxiv preprint*, abs/1910.10683.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *Journal of Machine Learning Research*.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using Abstract Meaning Representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021. Structural adapters in pretrained language models for amr-to-text generation. *Arxiv preprint*, abs/2103.09120.

R. H. Richens. 1958. Interlingual Machine Translation. *The Computer Journal*, 1(3):144–147.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Gözde Gül Şahin and Eşref Adalı. 2018. Annotation of semantic roles for the turkish proposition bank. *Language Resources and Evaluation*, 52(3):673–706.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 12–21, Prague, Czech Republic. Association for Computational Linguistics.

Janaki Sheth, Young-Suk Lee, Ramón Fernandez Astudillo, Tahira Naseem, Radu Florian, Salim Roukos, and Todd Ward. 2021. Bootstrapping multilingual AMR with contextual word alignments. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 394–404, Online. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for Relation Extraction and Semantic Role Labeling. *arXiv preprint*.

Marco Antonio Sobrevilla Cabezudo and Thiago 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. Association for Computational Linguistics.

Haoyu Song, Wei-Nan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019a. Exploiting persona information for diverse generation of conversational responses. In *Proc. of IJCAI*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019b. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019c. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7987–7998, Online. Association for Computational Linguistics.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 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)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017a. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017b. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008a. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of CoNLL*.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008b. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. of NeurIPS*.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

Jens Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Timothy J. O'Gorman, Andrew Cowell, William Bruce Croft, Chu-Ren Huang, Jan Hajic, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *Künstliche Intell.*, 35:343–360.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *CoRR*, abs/1705.09980.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Proc. of NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017c. Attention is all you need. In *Proc. of NIPS 2017*, NIPS'17, page 6000–6010, Red Hook, NY, USA.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1503–1515, New Orleans, Louisiana. Association for Computational Linguistics.

Tianming Wang, Xiaojun Wan, and Shaowei Yao. 2020. Better amr-to-text generation with graph structure reconstruction. In *Proc. of IJCAI 2020*, pages 3919–3925.

Shira Wein and Nathan Schneider. 2021. Classifying divergences in cross-lingual AMR pairs. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 56–65, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. In *Proc. of AAAI*.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving AMR parsing with sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2501–2511, Online. Association for Computational Linguistics.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2021. XLPT-AMR: Cross-lingual pre-training via multi-task learning for zero-shot AMR parsing and text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 896–907, Online. Association for Computational Linguistics.

Nianwen Xue, Ondřej Bojar, Jan Hajič, Martha Palmer, Zdeňka Urešová, and Xiuhong 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. European Language Resources Association (ELRA).

Nianwen Xue and Martha Palmer. 2003. Annotating the propositions in the Penn Chinese treebank. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 47–54, Sapporo, Japan. Association for Computational Linguistics.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7145–7154, Online. Association for Computational Linguistics.

Mark Yatskar, Luke S. Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual Semantic Role Labeling for image understanding. In *Proc. of CVPR*.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative Question Answering. In *Proc. of IJCAI*.

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

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. 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 (EMNLP-IJCNLP)*, pages 3786–3798, Hong Kong, China. Association for Computational Linguistics.

Yan Zhang, Zhijiang Guo, Zhiyang Teng, Wei Lu, Shay B. Cohen, Zuozhu Liu, and Lidong Bing. 2020. Lightweight, dynamic graph convolutional networks for AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 2162–2172, Online. Association for Computational Linguistics.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 732–741, Online. Association for Computational Linguistics.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, and Radu Florian. 2021. AMR parsing with action-pointer transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5585–5598, Online. Association for Computational Linguistics.

Junru Zhou, Zuchao Li, and Hai Zhao. 2020a. Parsing all: Syntax and semantics, dependencies and spans. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4438–4449, Online. Association for Computational Linguistics.

Qiji Zhou, Yue Zhang, Donghong Ji, and Hao Tang. 2020b. AMR parsing with latent structural information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4306–4319, Online. Association for Computational Linguistics.

Huaiyu Zhu, Yunyao Li, and Laura Chiticariu. 2019a. Towards universal semantic representation. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 177–181, Florence, Italy. Association for Computational Linguistics.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019b. 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 (EMNLP-IJCNLP)*, pages 5459–5468, Hong Kong, China. Association for Computational Linguistics.

Appendix A

SPRING Demonstration and Examples

A.1 SPRING Online Services Evaluation

For the purposes of this SPRING demo, we examine different variants of SPRING to ensure: i) high performance, ii) high generalizability across domains, and iii) efficient and light SPRING Online Services.

Datasets. To deal with i) and ii), we perform experiments with the AMR 3.0 (LDC2020T02[1]) benchmark – currently the largest AMR-annotated corpus which includes and corrects both of its previous inferior-sized versions, i.e., AMR 2.0 and AMR 1.0. In addition to this, motivated by AMR-based approaches in biomedical applications Rao et al. [2017]; Bonial et al. [2020b], we jointly train and evaluate SPRING in the Bio-AMR[2] corpus May and Priyadarshi [2017] as well.

Systems. While Bevilacqua et al. [2021a] train one specular model for each of the AMR tasks (henceforth SPRING$_{\text{uni}}$, denoting unidirectional), to satisfy the point iii) above, we train a version of SPRING that handles both AMR parsing and generation with the same model (henceforth SPRING$_{\text{bi}}$, denoting bidirectional). This allows us to load into memory only one model to perform both tasks, thus decreasing the potential overload of the server where

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

[2] amr.isi.edu/download.html

Parsing	Lyu et al. [2020]	75.8
	Zhou et al. [2021]	81.2
	SPRING Bevilacqua et al. [2021a]	**83.0**
Generation	Zhang et al. [2020]	34.3
	T5 Fine-Tune Ribeiro et al. [2021]	41.6
	STRUCTADAPT-RGCN Ribeiro et al. [2021]	**48.0**
	SPRING Bevilacqua et al. [2021a]	44.9

Table A.1. Comparison with literature on AMR 3.0.

the demo resides, as well as enabling lower memory footprint for users employing SPRING with our Python code. To train SPRING variants, we employ the same hyperparameters as in Bevilacqua et al. [2021a]. In addition, we summarize the state-of-the-art systems on AMR 3.0.

Results. We report Smatch Cai and Knight [2013] and BLEU Papineni et al. [2002] scores for AMR parsing and generation, respectively. In Table A.1 we summarize the performances of recent systems in the literature on the AMR 3.0 parsing and generation tasks. In *parsing*, SPRING achieves the highest results across the board. In fact, we note that Zhou et al. [2021] was published after Bevilacqua et al. [2021a], yet SPRING remains the best-performing parser in the literature to date. In *generation*, instead, SPRING attains considerably higher results than Zhang et al. [2020] and T5 Fine-Tune Ribeiro et al. [2021] models. In fact, while the latter has a comparable architecture to that of SPRING due to its use of the pretrained sequence-to-sequence T5 model Raffel et al. [2019], SPRING nevertheless outperforms it by 3.3 BLEU points. SPRING obtains lower results than the recent STRUCTADAPT-RGCN Ribeiro et al. [2021] model, which, however, achieved those results at the expense of a more complex architecture with a higher number of parameters than SPRING. In Table A.2 we report the performance of SPRING variants, i.e., SPRING$_{\text{uni}}$ and SPRING$_{\text{bi}}$, trained on AMR 3.0 or on the concatenation of Bio-AMR and AMR 3.0 (Bio+AMR 3.0) and when evaluated in development and test splits of each. Notice that the results of SPRING$_{\text{uni}}$ in AMR 3.0 parsing are different from those reported in Table A.1, since here we do not perform Entity Linking in postprocessing for the purpose of simplicity. Firstly, SPRING models trained on Bio+AMR 3.0 achieve the highest results overall. Then, SPRING$_{\text{bi}}$ performs on

			AMR 3.0		Bio-AMR	
		Train dataset	Dev	Test	Dev	Test
Parsing	SPRING$_{\text{uni}}$	AMR 3.0	83.9	82.6	60.6	60.6
	SPRING$_{\text{bi}}$	AMR 3.0	83.6	82.3	60.5	59.2
	SPRING$_{\text{uni}}$	Bio+AMR 3.0	83.9	82.5	**80.0**	80.1
	SPRING$_{\text{bi}}$	Bio+AMR 3.0	**84.1**	**82.7**	79.5	**80.2**
Generation	SPRING$_{\text{uni}}$	AMR 3.0	45.0	44.9	22.9	19.4
	SPRING$_{\text{bi}}$	AMR 3.0	43.9	44.5	21.1	17.1
	SPRING$_{\text{uni}}$	Bio+AMR 3.0	**45.3**	**45.7**	**39.5**	**43.5**
	SPRING$_{\text{bi}}$	Bio+AMR 3.0	44.3	45.0	38.5	42.0

Table A.2. SPRING variants in AMR 3.0 and Bio-AMR.

a par with or slightly worse than SPRING$_{\text{uni}}$ in parsing and generation, respectively. We choose the best model for the SPRING Online Services based on the Smatch score on the development set of AMR 3.0, i.e, SPRING$_{\text{bi}}$ trained on Bio+AMR 3.0 for both parsing and generation jointly. This model allows for the achievement of all the goals we set at the beginning of this Section: performance, generalizability and efficiency [3].

A.2 From Parsing to Generation

In this Section we report a few examples of parsing and generation obtained by running our DFS-based models trained on AMR 2.0. We collect some a few excerpts from the prompts shown by Radford et al. [2019], parse them into graphs and generate a sentence from the parsed graph. Results are shown in Table A.3. We also include them (with the graph linearization indented for better readability) in the `samples.txt` file in the provided code. As one can see, the generated sentenced from the parsed graphs preserve the meaning of the original sentence, thus demonstrating the high quality of the outputs from both SPRING$^{\text{DFS}}$ parser and generator. Note that the sentences are quite diverse, including things that are probably not present in the training data of SPRING$^{\text{DFS}}$ – thus confirming its generalizability power.

[3] We release the additional model checkpoints to be used with the original SPRING Python code, available at https://github.com/SapienzaNLP/spring

Original	→ Parsed graph	→ Generated Sentence
In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains.	(z0 / discover-01 :ARG0 (z1 / scientist) :ARG1 (z2 / herd :consist-of (z3 / unicorn) :ARG0-of (z4 / live-01 :location (z5 / valley :mod (z6 / remote) :ARG1-of (z7 / explore-01 **:polarity** - :time (z8 / previous)) :location (z9 / mountain :wiki "Andes" :name (z10 / name :op1 "Andes" :op2 "Mountains"))))) :ARG0-of (z11 / shock-01))	Scientists were shocked to discover a herd of unicorns living in a remote valley inaccessible in the Andes Mountains.
Emily loves mint chocolate cake, but she requires that it be paired with mini chocolate chips, so I threw some of those in between the layers.	(z0 / love-01 :ARG0 (z1 / person :wiki - :name (z2 / name :op1 "Emily")) :ARG1 (z3 / cake :consist-of (z4 / chocolate :mod (z5 / mint))) :concession-of (z6 / require-01 :ARG0 z1 :ARG1 (z7 / pair-01 :ARG1 z3 :ARG2 (z8 / chip :consist-of (z9 / chocolate :mod (z10 / mini))))) :ARG0-of (z11 / cause-01 :ARG1 (z12 / throw-01 :ARG0 (z13 / i) :ARG1 (z14 / some :ARG1-of (z15 / include-91 :ARG2 z3)) :ARG2 (z16 / between :op1 (z17 / layer)))))	Emily loves chocolate cake, but it requires it to be paired with mini chocolate chips, so I threw some of them in between the layers.
Prehistoric man sketched an incredible array of prehistoric beasts on the rough limestone walls of a cave in modern day France 36,000 years ago.	(z0 / draw-01 :ARG0 (z1 / man :mod (z2 / prehistoric)) :ARG1 (z3 / array :mod (z4 / incredible) :consist-of (z5 / beast :mod (z6 / prehistoric))) :location (z7 / wall :consist-of (z8 / limestone) :ARG1-of (z9 / rough-04) :part-of (z10 / cave :location (z11 / country :wiki "France" :name (z12 / name :op1 "France") :time (z13 / day :ARG1-of (z14 / modern-02))))) :time (z15 / before :op1 (z16 / now) :quant (z17 / temporal-quantity :quant 36000 :unit (z18 / year))))	36,000 years ago, prehistoric men drew an incredible array of prehistoric beasts on a rough limestone wall of a cave in modern-day France.
Corporal Michael P. Goeldin was an unskilled laborer from Ireland when he enlisted in Company A in November 1860.	(z0 / person :ARG0-of (z1 / labor-01 :manner (z2 / skill **:polarity** -)) :domain (z3 / person :wiki - :name (z4 / name :op1 "Michael" :op2 "P." :op3 "Goeldin") :ARG0-of (z5 / have-org-role-91 :ARG2 (z6 / corporal))) :mod (z7 / country :wiki "Ireland" :name (z8 / name :op1 "Ireland")) :time (z9 / enlist-01 :ARG1 z3 :ARG2 (z10 / military :wiki - :name (z11 / name :op1 "Company" :op2 "A")) :time (z12 / date-entity :year 1860 :month 11)))	When Michael P. Goeldin enlisted in Company A in November, 1860, he was an Irish labourer with no skills.

Table A.3. Original sentence (left); DFS-based parser output (middle); DFS-based generator output when the input is the parsed graph (right).

Appendix B

XL-AMR Details

B.1 OpusMT Translation Models

For the translation and back-translation steps of GOLDAMR-SILVERTRNS data creation approach, we use the pretrained models[1] from the huggingface transformers library[2] listed in Table B.1.

Source	Target	Model
German	English	`Helsinki-NLP/opus-mt-de-en`
Italian	English	`Helsinki-NLP/opus-mt-it-en`
Spanish	English	`Helsinki-NLP/opus-mt-ROMANCE-en`
English	German	`Helsinki-NLP/opus-mt-en-de`
English	Italian	`Helsinki-NLP/opus-mt-en-it`
English	Spanish	`Helsinki-NLP/opus-mt-en-ROMANCE`

Table B.1. OpusMT translation models.

B.2 Model Hyperparameters

The input features for all the models include: i) fixed mBERT[3] [Devlin et al., 2019] as contextual embeddings (dim = 768), ii) ConceptNet Numberbatch 9.08[4] [Speer et al., 2017b] multilingual static word embeddings (dim = 300) which we set as trainable except in $\emptyset$-shot

[1] 6-layer Transformer-based models [Vaswani et al., 2017b].

[2] huggingface.co/transformers/model_doc/marian.html

[3] `bert-base-multilingual-cased`: a contextualized embedding for a token is calculated as the average pooling of its subtoken embeddings.

[4] github.com/commonsense/conceptnet-numberbatch

models, iii) trainable PoS embeddings (dim = 100) where we use the universal PoS-tags set by Petrov et al. [2012], iv) trainable anonymization indicator embeddings (dim = 50), v) trainable character-level embeddings (dim = 100), i.e., CharCNN [Kim et al., 2016].

The encoder and decoder of the node prediction module are composed of 2 layers of 512 and 1024 LSTM units each, respectively. All the models are trained using Adam optimizer [Kingma and Ba, 2015] with learning rate 0.001, for 120 epochs and the best model hyperparameters are chosen on the basis of development set accuracy. The models are trained using 1 GeForce GTX TITAN X GPU. The full training lasts around 48 hours for models trained in the largest dataset XL-AMR^{trans+} ($\sim$84M trainable parameters) and XL-AMR^{par+} ($\sim$86M trainable parameters). At prediction time we set the size of beam search to 5.