Maximum Bayes Smatch Ensemble Distillation for AMR Parsing

Young-Suk Lee†, Ramón Fernandez Astudillo†, Thanh Lam Hoang‡,
Tahira Naseem†, Radu Florian†, Salim Roukos†
{ysuklee,tanseem,raduf,roukos}@us.ibm.com
ramon.astudillo@ibm.com
t.l.hoang@ie.ibm.com
IBM Research AI†, IBM Research - Ireland‡

Abstract

AMR parsing has experienced an unprecendented increase in performance in the last three years, due to a mixture of effects including architecture improvements and transfer learning. Self-learning techniques have also played a role in pushing performance forward. However, for most recent high performant parsers, the effect of self-learning and silver data augmentation seems to be fading. In this paper we propose to overcome this diminishing returns of silver data by combining Smatch-based ensembling techniques with ensemble distillation. In an extensive experimental setup, we push single model English parser performance to a new state-of-the-art, 85.9 (AMR2.0) and 84.3 (AMR3.0), and return to substantial gains from silver data augmentation. We also attain a new state-of-the-art for cross-lingual AMR parsing for Chinese, German, Italian and Spanish. Finally we explore the impact of the proposed technique on domain adaptation, and show that it can produce gains rivaling those of human annotated data for QALD-9 and achieve a new state-of-the-art for BioAMR.

1 Introduction

Adoption of the Transformer architecture (Vaswani et al., 2017) for Abstract Meaning Representation (AMR) parsing (Cai and Lam, 2020; Fernandez Astudillo et al., 2020) as well as pretrained language models (Bevilacqua et al., 2021; Zhou et al., 2021b; Bai et al., 2022) have enabled an improvement of above 10 Smatch points (Cai and Knight, 2013), the standard metric, in the last two years.

Data augmentation techniques have also shown great success in pushing the state-of-the-art of AMR parsing forward. These include generating silver AMR annotations with a trained parser (Konstas et al., 2017; van Noord and Bos, 2017), with multitask pre-training and fine-tuning (Xu et al., 2020) as well as combining AMR to source text and silver AMR generation (Lee et al., 2020) and stacked pre-training of silver data from different models – from low performance to high performance silver data (Xia et al., 2021). However, the latest BART-based state-of-the-art parsers, have shown diminishing returns for data augmentation. Both SPRING (Bevilacqua et al., 2021) and Structured-BART (Zhou et al., 2021b) gain a mere 0.5 Smatch from self-learning, compared with over 1 point gains of the previous, less performant, models. Since performance scores are already above where inter annotator agreement (IAA) is assumed to be, i.e. 83 for newswire and 79 for web text reported in (Banarescu et al., 2013), one possible explanation is that we are reaching some unavoidable performance plateau.

In this work we show that we can achieve significant performance gains close to 2 Smatch point with the newly proposed data augmentation technique, contrary to the results from the previous state-of-the-art systems. The main contributions of this paper are as follows:

- We propose to combine Smatch-based model ensembling (Barzdins and Gosko, 2016; Hoang et al., 2021) and ensemble distillation (Hinton et al., 2015) of heterogeneous parsers to produce high quality silver data.
- We offer a Bayesian ensemble interpretation of this technique as alternative to views such as Minimum Bayes Risk decoding (Goel and Byrne, 2000) and name the technique Maximum Bayes Smatch Ensemble (MBSE).
- Applied to English monolingual parsing, MBSE distillation yields a new single system state-of-the-art (SoTA) on AMR2.0 (85.9) and AMR3.0 (84.3) test sets.
- Trained with Structured-mBART[1], it yields new SoTA for Chinese (63.0), German (73.7),

[1] https://github.com/IBM/transition-amr-parser/tree/structured-mbart

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5379 - 5392
July 10-15, 2022 ©2022 Association for Computational Linguistics

Italian (76.1) and Spanish (77.1) cross-lingual parsing.

- Applied to domain adaptation, MBSE distillation achieves the performance comparable to human annotations of QALD-9 training data and achieves new SoTA on BioAMR test set.

- We release QALD-9-AMR treebank[2] at, which comprises 408 training and 150 test sentences.

2 Maximum Bayes Smatch Ensemble

Ensemble distillation (Hinton et al., 2015) integrates knowledge of different teacher models into a student model. For sequence to sequence models, e.g. machine translation, it is possible to ensemble models by combining probabilities of words given context at each time step (Kim and Rush, 2016; Freitag et al., 2017). Syntactic and semantic parsers model a distribution over graphs that is harder to integrate across teacher models in an optimal way. For particular cases like dependency parsing, it is possible to ensemble teachers based on the notion of edge attachment (Kuncoro et al., 2016), which is related to the usual evaluation metric, Label Attachment Score (LAS). However, AMR graphs are quite complex and not explicitly aligned to words. The standard Smatch (Cai and Knight, 2013) metric approximates the NP-Complete problem of aligning nodes across graphs with a hill climbing algorithm. This illustrates the difficulty of achieving consensus across teachers for AMR ensembling.

Prior work ensembling AMR graphs has leveraged Smatch directly or its hill climbing strategy for ensembling. The ensemble in (Barzdins and Gosko, 2016) selects, among a number of candidate AMRs, the one that has the largest average Smatch with respect to all sampled AMRs. The ensemble in (Hoang et al., 2021), modifies the candidate AMRs to increase consensus as measured by coverage. Then it selects from the union of original and modified graphs for the one with highest coverage or largest average Smatch. One possible intepretation of both techniques is that of Minimum Bayes Risk (MBR) decoding, a well established method in Automatic Speech Recognition (ASR) (Goel and Byrne, 2000) and Machine Translation (MT) (Kumar and Byrne, 2004). Assuming that we have a model predicting a graph from an input sentence $p(g \mid w)$, normal decoding entails searching among model outputs g for the one that has the highest likelihood according to the model $p(g \mid w)$. MBR searches instead for the model output that minimizes the risk with respect to the distribution of possible human (gold) outputs for a given input

$$\hat{g} = \arg\min_{g}\{E_{p(g^h|w)}\{R(g, g^h)\}\}$$

where $p(g^h \mid w)$ is the distribution of correct human outputs, e.g. given by multiple annotators, and R is a risk function that measures how severe deviations from g^h are. In this case risk would be minus Smatch. Since in practice $p(g^h \mid w)$ is not available, MBR takes often the strong assumption of replacing $p(g^h \mid w)$ by the model distribution itself $p(g \mid w)$.

Here we suggest another Bayesian interpretation, that requires less strong assumptions than MBR, a Bayesian model ensemble (Wilson and Izmailov, 2020). Indeed techniques above can be seen as solving

$$\hat{g} = \arg\max_{g \in \mathcal{G}}\{E_{p(\mathcal{M}|\mathcal{D})}\{\text{Smatch}(g, \tilde{g}_{\mathcal{M}})\}\}$$

where $p(\mathcal{M} \mid \mathcal{D})$ is the distribution of models $\mathcal{M}$ given training data $\mathcal{D}$, approximated by a sample average of models of different architectures or different random seeds, and

$$\tilde{g}_{\mathcal{M}} = \text{post}\left(\arg\max_{y}\left\{\prod_{t=1}^{|y|} p_{\mathcal{M}}(y_t \mid y_{<t}, w)\right\}\right)$$

is the output of a conventional decoding process for each parser prediction distribution $p_{\mathcal{M}}$, including post-processing $\text{post}()$. This process differs across models indexed by $\mathcal{M}$, for example y can be transition actions or linearized graphs and $\text{post}()$ running the state-machine or linearized graph post-processing[3]. $\mathcal{G}$ is the space of candidate graphs, which in Barzdins and Gosko (2016) are the AMRs resulting from decoding each sample from $p(\mathcal{M} \mid \mathcal{D})$ and in Hoang et al. (2021) are those same graphs plus the modified pivot graphs. There is in principle no restriction on how to build the set $\mathcal{G}$. *Decoding* a graph $g \in \mathcal{G}$ means here selecting the member of that set maximizing the expected Smatch and is different from each parser's decoding process.

[2] https://github.com/IBM/AMR-annotations

[3] We consider only auto-regressive models in this work but this approach could also encompass e.g. graph-based parsers.

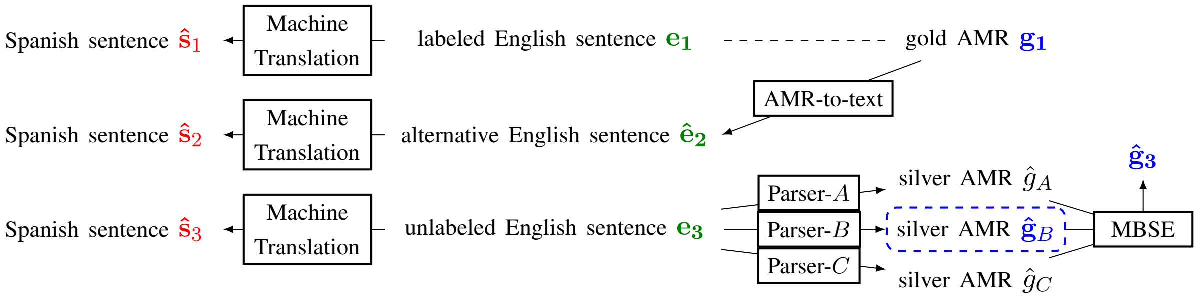

Figure 1: Data augmentation framework: Given a labeled example in English (e_1, g_1), we use an AMR-to-text generation system to generate an alternative input text $\hat{e}_2$ for g_1 following (Lee et al., 2020). Given a sentence e_3, and various state-of-the-art off-the-shelf parser outputs (A, B, C), Maximum Bayes Smatch Ensemble (MBSE) produces a single annotation for each input sentence by selecting from existing AMRs or their modified versions. MBSE is only applied to unlabeled English sentences to produce $\hat{g}_3$. Following (Damonte and Cohen, 2018), we translate the English sentences to e.g. Spanish, to yield new training samples $(\hat{s}_1, g_1)$, $(\hat{s}_2, g_1)$, $(\hat{s}_3, \hat{g}_3)$ to train a Spanish cross-lingual parser. We use the English pairs (e_1, g_1), $(\hat{e}_2, g_1)$, $(e_3, \hat{g}_3)$ to train an English parser.

If we replace $\mathrm{Smatch}()$ by an indicator function on the decoding outputs $1_{g=\tilde{g}_\mathcal{M}}$, then

$$\hat{g} = \arg\max_{g \in \mathcal{G}} \{E_{p(\mathcal{M}|\mathcal{D})}\{1_{g=\tilde{g}_\mathcal{M}}\}\}$$

recovers majority voting of AMR graphs. Since the space of graphs is exponentially large on the input size, this would be too sparse to attain meaningful vote counts. The propagation of the uncertainty in $p(\mathcal{M} \mid \mathcal{D})$ through the $\mathrm{Smatch}()$ transformation both solves the sparsity problem, and allows optimization on a space that is better related to the target metric. The method will be henceforth described here as Maximum Bayes Smatch Ensemble distillation (MBSE distillation).

In what follows, we will consider three versions for ensembling, the Smatch version of Hoang et al. (2021) (graphene-Smatch), the average-Smatch selection of Barzdins and Gosko (2016), and a greedy version of Barzdins and Gosko (2016) where we select the two highest Smatch AMRs and from that pair, keep the graph with the highest Smatch with respect to the remaining graphs (greedy-select). The greedy-select algorithm is given in Algorithm 1 of Appendix A and performs similarly to the average-Smatch of Barzdins and Gosko (2016).

3 Silver Training Strategy

We now describe the AMR silver training strategy proposed in this work. This strategy creates high quality English and cross-lingual AMR annotations for unlabeled data with MBSE and alternative input sentences of gold AMRs via AMR-to-text.

As depicted in Fig. 1, we start with 1) a set of gold-labeled (English sentence, AMR) pairs, 2) a set of unlabeled English sentences and 3) pre-trained English-to-foreign language Machine Translation systems. Assuming N off-the-shelf AMR parsers, we train each of the N parsers using the gold data with their respective training procedure. More than one random seed may be trained for some parsers, leading to more than N AMR parses for each input sentence.

After the parsers have been trained, we use them to parse the unlabeled English text as in Konstas et al. (2017). Interpreting the set of trained models as samples of the model distribution, we apply the MBSE distillation methods described in Section. 2. We apply all variations of the MBSE algorithms including graphene-Smatch, greedy-select and average-Smatch algorithms.

For English parsers, the MBSE distilled AMR annotations are added to the human-annotated gold treebanks for enhanced model training. For cross-lingual parsers, we translate all English input sentences to the target foreign languages and train respective cross-lingual parsers with pairs of (Foreign language input sentences, AMR graphs in English), following (Damonte and Cohen, 2018).

Following Lee et al. (2020), we also apply an AMR-to-text model (Mager et al., 2020; Ribeiro et al., 2021; Bevilacqua et al., 2021) to generate additional sentences for human-annotated AMR. We filter out the generated texts if they are too similar (BLEU > 0.9) or too dissimilar (BLEU < 0.1) to the original input texts, as measured by BLEU (Papineni et al., 2002). AMR-to-text generation[4] is used for cross-lingual AMR parser training only.

[4] We use https://github.com/SapienzaNLP/spring.

For Standard Experiments				For Domain Adaptation			
Dataset	**Split**	**Sents**	**Tokens**	**Dataset**	**Split**	**Sents**	**Tokens**
AMR2.0	Train	36,521	653K	QALD-9-AMR (*new*)	Train	408	3,475
	Test	1,371	30K		Test	150	1,441
	Dev.	1,368	29K				
AMR3.0	Train	55,635	1M	Bio AMR	Train	5,452	231K
	Test	1,898	39K		Test	500	22K
	Dev.	1,722	37K				
				LP	Test	1,562	21K
PropBank	silver^1	20K	386K	SQuAD2.0-Q	silver^q	135K	1.5M
SQuAD2.0-C	silver^1	70K	2M	BioNLP-ST-2011	silver^b	15K	460K
Ontonotes5.0	silver^2	59K	1.1M	CRAFT	silver^b	27K	740K
WikiText-103	silver^3	70K	2M	PubMed	silver^b	26K	750K

Table 1: Corpus statistics for the standard benchmark experiments on AMR2.0 and AMR3.0 test sets (left) and domain adaptation experiments (right). Silver indicates the unlabeled data for silver training.

4 Experimental Setup

4.1 Corpus Statistics and QALD-9-AMR

Table 1 details the corpora considered for the standard benchmark experiments on AMR2.0 and AMR3.0 test sets (lef) and out-of-domain data used for domain adaptation experiments (right). Silver indicates the unlabeled data for silver AMR acquisition. SQuAD2.0-Q(uestions) are for QALD-9 (silver^q) and PubMed, BioNLP-2011 (Kim et al., 2011) and CRAFT (Cohen et al., 2017) for BioAMR (silver^b).

Since there were no human annotations of QALD-9 corpus, we created QALD-9-AMR treebank. QALD-9 training/test data have been annotated by 3 skilled resident human annotators with experience in AMR annotations over a year. Each of the annotators annotated both the train and test data sets, followed by cross validation by each other. The final annotations were adjudicated by the most experienced annotator. Inter-annotator agreement (IAA) rate on a subset of 158 training sentences is over 95% in Smatch. The data is made publicly available under an Apache2 license.

4.2 Parsing Models

We use 4 off-the-shelf AMR parsers to parse unannotated raw texts. We train the parsers following their standard configurations.

APT (Zhou et al., 2021a)[5] is a transition-based parser that combines hard attention over sentences with a target side action pointer mechanism to decouple source tokens from node representations and address alignments. Cross-attention of all decoder layers is used for action-source alignment.

SPRING Bevilacqua et al. (2021)[6] fine-tunes BART (Lewis et al., 2020) to predict linearized AMR graphs, avoiding complex pipelines.

Structured-BART Zhou et al. (2021b)[7] models the transition-based parser state within a pre-trained BART architecture, outperforming SPRING. This is the main parser for our work.

AMRBART Bai et al. (2022)[8] improves the structure awareness of pre-trained BART over AMR graphs by introducing node/edge denoising and sub-graph denoising tasks, for graph-to-graph pre-training, achieving significant improvement over previous BART-based systems.

4.3 Structured-mBART

For cross-lingual AMR parsing, we adapt Structured-BART by replacing the pretrained BART with mBART of (Liu et al., 2020), henceforth Structured-mBART. The codebase is made publicly available under an Apache2 license. Structured-mBART diverges from Structured-BART mainly in input processing and vocabulary:

- For task vocabulary, Structured-mBART includes ~250K sentencepiece tokens of (Kudo, 2018) including 25 language tags, e.g. es_XX, whereas Structured-BART includes ~50K BPE tokens of (Sennrich et al., 2016).

- We append the source language tag to the end

[5] github.com/IBM/transition-amr-parser/tree/action-pointer, Apache2 License

[6] github.com/SapienzaNLP/spring, CC BY-NC-SA 4.0

[7] github.com/IBM/transition-amr-parser, Apache2 License

[8] https://github.com/muyeby/AMRBART, MIT License

Models	AMR2.0	AMR3.0	Q9AMR	LP	BioAMR
APT (Zhou et al., 2021a)	83.0	81.1	83.7	79.0	55.2
Structured-BART (Zhou et al., 2021b)	84.6	83.1	87.7	81.0	62.4
SPRING$_1$ (Bevilacqua et al., 2021)	84.2	83.2	87.7	81.3	61.6
SPRING$_2$ (Bevilacqua et al., 2021)	83.8	82.9	86.4	81.0	60.5
AMRBART (Bai et al., 2022)	85.4	84.2	88.0	82.3	63.4
aver.-Smatch (A) (Barzdins and Gosko, 2016)	86.2	84.9	89.0	82.9	64.1
graphene-Smatch (P) (Hoang et al., 2021)	**86.7**	**85.4**	**89.3**	**83.1**	**65.8**
greedy-select (G)	85.9	84.8	88.8	82.8	63.9

Table 2: English parsing performance in Smatch in general domain and domain adaptation for recent state-of-the-art systems (top). Performance in Smatch for the ensemble of all systems using different Smatch-based ensembling techniques (bottom). SPRING$_1$ and SPRING$_2$ are 2 random seeds of the same model. Highest scores are **boldfaced**.

of each input sentence without specifying the target language tag for Structured-mBART.

- For Structured-mBART, we set the learning rate to $3\text{e}{-}5$, cf. $1\text{e}{-}4$ of Structured-BART, and move the layer normalization to the beginning of each transformer block.

We obtain contextualized embeddings from the pre-trained mBART for multilingual input sentence representations. For target action sequences, we map the sentencepiece tokens to the corresponding target token, by averaging all values from the sentencepiece tokens corresponding to the target token. For German, Italian and Spanish input texts, we apply the tokenizer from JAMR parser[9] before sentencepiece tokenization. For Chinese, we directly apply the sentencepiece tokenizer.

5 Results

To explore the effect of the proposed MBSE distillation and training strategy, we consider an extensive experimental setup including standard English benchmarks (Section 5.1), cross-lingual benchmarks (Section 5.2) and out of domain data sets (Section 5.3).[10] For model training and selection details, see Appendix B and Appendix C.

We first provide the performance evaluation of each ensembling technique used in MBSE in Table 2 to demonstrate the effectiveness of the ensemble techniques by themselves. We test the algorithm on the standard test data sets from AMR2.0 and AMR3.0 and three out-of-domain data sets, Q9AMR (QALD-9-AMR), LP (Little Prince) and BioAMR in Table 1. We consider here only standard English AMR parsing. As expected, all MBSE algorithms, average-Smatch, graphene-Smatch and greedy-select, improve individual models by large margins. Note that while the ensembles outperform single model state-of-the-art by a large margin, the use of heterogeneous ensembles of models is computationally prohibitive in practice, both due to the cost of running different models but also the ensembling techniques.

5.1 English AMR Parsing

As displayed in Table 1, we consider the standard AMR2.0 (LDC2017T10) and AMR3.0 (LDC2020T02) treebank as gold data. For ensemble distillation, we use the data sets denoted by silver^1 for comparison with previous work, and silver^2 and silver^3 to investigate the impact of unlabeled corpus size on model performance. For silver^1, we use all sentence examples in PropBank (LDC2004T14). From SQuAD2.0-C(ontexts)[11] we filter out the ~92K sentences, removing bad utf8 encoding (~7K) and ill-formed disconnected graphs produced by APT (~15K). Silver^2 comprises Ontonotes5.0 (LDC2013T19) and silver^3 WikiText-103[12]

The results are shown in Table 3. The lower part of the table (denoted by **Ours**) compares the performances of Structured-BART in various silver data augmentation setups including our proposed MBSE distillation. With the same unlabeled corpus silver^1, greedy-select distillation improves 1.0 Smatch point on AMR2.0 (84.2 vs. 85.2) and 1.5 Smatch point on AMR3.0 (82.0 vs. 83.5) over the

[9] https://github.com/jflanigan/jamr

[10] We also applied the technique to APT, observing similar performance gains when using MBSE distillation.

[11] https://rajpurkar.github.io/SQuAD-explorer/

[12] https://www.salesforce.com/products/einstein/ai-research/the-wikitext-dependency-language-modeling-dataset/

Models	silver	AMR2.0		AMR3.0	
Naseem et al. (2019)		75.5		-	
Zhang et al. (2019a)		76.3±0.1		-	
Zhang et al. (2019b)		77.0±0.1		-	
Cai and Lam (2020)		80.2		-	
Fernandez Astudillo et al. (2020)		80.2±0.0		-	
Lyu et al. (2020)		-		75.8	
Lee et al. (2020)	85K	81.3±0.0		-	
Xu et al. (2020)	14M	81.4		-	
Bevilacqua et al. (2021)	200K	84.5		83.0	
Zhou et al. (2021a)	70K	82.6±0.1		80.3	
Xia et al. (2021)	1.8M	84.2		-	
Bai et al. (2022)	200K	**85.4**		**84.2**	
Zhou et al. (2021b)		sep-voc	joint-voc	sep-voc	joint-voc
Structured-BART-baseline		84.0±0.1	84.2±0.1	82.3±0.0	82.0±0.0
+ self-trained silver^{1}	90K	-	84.7±0.1	82.7±0.1	82.6±0.0
+ self-trained silver^{1} + ensemble dec.	90K	-	84.9	83.1	-
Ours below (Struct-BART)		sep-voc	joint-voc	sep-voc	joint-voc
+ SPRING silver^{1}	90K	84.8±0.1	84.8±0.0	83.0±0.0	83.2±0.1
+ SPRING + self-trained silver^{1} (50:50)	90K	84.8±0.1	84.7±0.0	83.0±0.0	83.2±0.1
Ensemble-4 distillation (APT + Structured-BART + SPRING$_1$ + SPRING$_2$)					
+ MBSE-P silver^{1}	90K	85.1±0.1	85.1±0.1	83.2±0.1	83.5±0.1
+ MBSE-G silver^{1}	90K	85.0±0.0	85.2±0.1	83.4±0.0	83.5±0.0
+ MBSE-G silver^{1+2}	149K	85.3±0.1	85.4±0.1	83.6±0.1	83.7±0.1
+ MBSE-G siver^{1+2+3}	219K	85.3±0.1	85.5±0.1	83.7±0.0	83.9±0.0
+ MBSE-G silver^{1+2+3} + ensemble dec.	219K	**85.6**	**85.7**	**84.0**	**84.2**
Ensemble-5 distillation (APT + Structured-BART + SPRING$_1$ + SPRING$_2$ + AMRBART)					
+ MBSE-A silver^{1}	90K		85.3±0.1		83.6±0.1
+ MBSE-A silver^{1+2}	149K		85.5±0.0		84.0±0.0
+ MBSE-A silver^{1+2+3}	219K		85.7±0.0		84.1±0.0
+ MBSE-A silver^{1+2+3} + ensemble dec.	219K		**85.9**		**84.3**

Table 3: Smatch scores on AMR2.0 and AMR3.0 test data. Upper rows display the performances of recent published works. Structured-BART results in (Zhou et al., 2021b) are shown in middle rows. Lower rows show Structured-BART performances with various silver data augmentations. sep-voc denotes separate vocabulary and joint-voc, joint vocabulary. The numbers prefixed by ± indicate the standard deviation of Smatch scores across 3 seeds.

Structured-BART baselines. Graphene-Smatch distillation performs similarly to greedy-select one.

To isolate the effect of ensembling, we provide two additional baselines: 1) silver obtained from SPRING, which is expected to have complementary information to self-trained silver, and 2) an equal mixture of SPRING and Structured-BART (random 50:50), which tests if the MBSE selection strategy bears any effect. MBSE distillation outperforms these two baselines by between 0.2 and 0.5 Smatch point, depending on the scenario, proving that MBSE selection has a clear positive effect.

We also investigate the impact of unlabeled corpus size on model performance by adding silver^{2} and silver^{3} to silver^{1}, i.e. silver^{1+2} and silver^{1+2+3}. We observe additional 0.3-0.4 improvement, complementary to the one obtainable with conventional ensemble decoding. This pushes the numbers to 85.7 and 84.2, setting a new SoTA for single system with 4 model ensemble (Ensemble-4) distillation. Note that using 5 model ensemble (Ensemble-5) distillation moves the Smatch scores even higher to 85.9 for AMR2.0 and 84.3 for AMR3.0.

5.2 Cross-lingual AMR Parsing

For cross-lingual AMR parsing, we consider the well known cross-lingual extension of AMR2.0 (Damonte and Cohen, 2018). Our cross-lingual

Models	LM	DE	ES	IT	ZH
Translate and Parse Pipelines					
Uhrig et al. (2021)		67.6	72.3	70.7	59.1
WLT+Structured-BART+MBSE-G silver^1	BART	73.9	76.5	76.1	63.7
WLT+Structured-BART+MBSE-A silver^{1+2+3}	BART	**74.6**	**77.1**	**76.7**	**64.0**
Cross-lingual Parsers					
Blloshmi et al. (2020)		53.0	58.0	58.1	43.1
Sheth et al. (2021) (85K silver AMR)	XLMR	62.7	67.9	67.4	–
Procopio et al. (2021) (5M parallel corpus)	mBART$_{mt}$	69.8	72.4	72.3	58.0
Cai et al. (2021b)		64.0	67.3	65.4	53.7
Xu et al. (2021)		70.5	71.8	70.8	–
Cai et al. (2021a) (320K silver AMR)	mBARTmmt	**73.1**	**75.9**	**75.4**	**61.9**
Ours below (with Structured-mBART)					
Structured-mBART-baseline	mBART	69.9±0.0	74.4±0.3	73.3±0.2	59.9±0.0
Ensemble-4 distillation (APT + Structured-BART + SPRING$_1$ + SPRING$_2$)					
+ MBSE-G silver^1	mBART	72.5±0.1	76.5±0.2	75.4±0.0	62.2±0.1
+ MBSE-G silver^1+AMR2Text	mBART	72.9±0.1	76.6±0.0	75.6±0.0	62.3±0.0
+ MBSE-G silver^1+AMR2Text + ens. dec.	mBART	73.2	76.9	75.7	62.7
Ensemble-5 distillation (APT + Structured-BART + SPRING$_1$ + SPRING$_2$ + AMRBART)					
+ MBSE-A silver^{1+2+3}	mBART	73.5±0.1	**77.1**±0.2	76.0±0.1	62.7±0.1
+ MBSE-A silver^{1+2+3} + ens. dec.	mBART	**73.7**	77.0	**76.1**	**63.0**

Table 4: Cross-lingual parser Smatch scores on AMR2.0 human translated test sets. mBART$_{mt}$ of Procopio et al. (2021) indicates the mBART model fine-tuned on both semantic parsing tasks and the MT data. mBARTmmt of Cai et al. (2021a) indicates an NMT model by (Tang et al., 2020), trained from mBART covering 50 languages. Shortnames: MBSE-G (greedy-selection), MBSE-A (average-Smatch) 'ens. dec.', ensemble decoding.

parsers are trained with Structured-mBART, always using separate vocabulary (sep-voc). Input sentences of the English training data are machine translated into the target languages with WLT[13] to generate cross-lingual parser training data.

Table 4 shows the results on the human translated AMR2.0 test set, following standard practices. We provide results for recently published cross-lingual AMR parsers and different silver training versions of Structured-mBART. Structured-mBART with 4 model ensemble (Ensemble-4) distillation with just silver^1 improves the Smatch score by 2.1 to 2.6 over the Structured-mBART baselines, outperforming very strong previous SoTA from (Cai et al., 2021a) on Chinese and Spanish and tied on Italian. Increasing the input sentence diversity via AMR-to-text generation and ensemble decoding further improve the system performances, attaining new cross-lingual SoTA on all four languages. Increasing the silver training data size to silver^{1+2+3} and using 5 model ensemble (Ensemble-5) push the numbers higher by 0.2-0.5 Smatch points.

(Uhrig et al., 2021) report that translate-and-parse pipelines outperform the conventional cross-lingual parsers, we thus include translate-and-parse from the combination of WLT and Structured-BART + MBSE distillation. This out-performs the cross-lingual parsers by 0.6-1.0 Smatch on all languages except for Spanish, when trained with the same MBSE avg.-Smatch silver^{1+2+3} data.

Comparing the fine-grained F1 scores for cross-lingual parsers with those for English, as shown in Table 5, we observe that cross-lingual parsers are particularly worse than English for negation. For instance, German negations are often realized as a compound, as in ***nicht****tarifäre* (non - tariff), which is aligned to the non-negated stem portion of the concept *tariff*, losing its negation meaning. We observe similar issues in English with prefixed negations such as ***un****happy,* ***in****adequate,* ***a****typical.*

5.3 Domain Adaptation

We use the AMR2.0 version of BioAMR (medical domain) as this has clearly defined partitions[14] and was used in Bevilacqua et al. (2021). We also use QALD-9-AMR, constructed from QALD-9 data[15] (Usbeck et al., 2018), a corpus of natural language

[13] https://www.ibm.com/cloud/watson-language-translator

[14] amr.isi.edu/download/2016-03-14/amr-release-training-bio.txt, amr-release-dev-bio.txt, amr-release-test-bio.txt

[15] https://github.com/ag-sc/QALD

Languages	Smatch	Unlabeled	NoWSD	Concepts	NER	Neg.	Wiki	Reentrant	SRL
EN-mono	**85.9**	**89.0**	**86.3**	**92**	**93**	**75**	**81**	**78**	**85**
DE-cross	73.7	77.9	73.8	75	89	***48***	79	61	68
ES-cross	77.1	81.4	77.5	81	89	***62***	79	67	74
IT-cross	76.1	80.4	76.3	79	90	***56***	78	65	73
ZH-cross	63.0	67.9	63.1	65	85	***35***	70	51	58
DE-pipeline	74.6	78.9	74.8	75	91	***51***	80	62	68
ES-pipeline	77.1	81.1	77.3	80	91	***61***	79	66	74
IT-pipeline	76.7	80.9	76.9	79	91	***58***	80	65	73
ZH-pipeline	64.0	68.9	64.0	66	86	***40***	74	51	59

Table 5: Fine-grained F1 scores on the AMR2.0 test set for EN (English), DE (German), ES (Spanish), IT (Italian) and ZH (Chinese). EN-mono denotes English mono-lingual parser, {DE,ES,IT,ZH}-cross, cross-lingual parsers and {DE,ES,IT,ZH}-pipeline, translate-and-parse pipeline.

questions for executable semantic parsing (Kapanipathi et al., 2021). Corpus statistics of the domain adaptation data is summarized in Table 1.

Table 6 shows the experimental results. Results for SPRING are taken from Bevilacqua et al. (2021). For each test set, we report the results under three different training conditions, all of which include either AMR2.0 or AMR3.0 treebank in the training data: (1) use only silver data with MBSE distillation, (2) use only domain gold sentences, (3) use both silver data and domain gold sentences. Since BioAMR is annotated in AMR2.0 style and QALD-9-AMR in AMR3.0 style, we use the corresponding Structured-BART models as indicated in the table.

As for BioAMR data, MBSE distillation (with both graphene-Smatch and greedy-select) on silver^b – comprising PubMed (LDC2008T20, LDC2008T21), BioNLP-ST-2011 and CRAFT – improves over the Structured-BART baseline by 6.5 Smatch point (60.4 vs. 66.9). However, adding just 201 domain gold sentences to AMR2.0 treebank results in 11.9 Smatch point improvement over the baseline (60.4 vs. 72.3). A close inspection shows that this is largely due to the NER score improvement, as shown in the column under **NER**, i.e. NER score 27.0 in Structured-BART (AMR2.0) vs. 68.0 after adding 201 domain gold sentences. The dramatic impact of NE coverage no longer holds when we double the domain gold sentences from 201 to 403. In fact, MBSE greedy-select silver^b + 201 domain gold sentences (75.8) is more effective than doubling the domain gold sentences (74.3). Finally, by combining MBSE distillation on silver^b with 5K domain gold sentences, the system achieves 81.3 Smatch, outperforming the previous SoTA by 1.4.

Training Data	**Smatch**	**NER**
BioAMR Evaluations		
SPRING$^{\text{DFS}}$	59.7	
SPRING$^{\text{DFS}}$+ silver	59.5	
SPRING$^{\text{DFS}}$ (In domain)	79.9	
Ours		
Struct-BART (AMR2.0)	60.4	27.0
+MBSE-G silver^1	63.2	31.0
+MBSE-G silver^b	66.9±0.2	30.0
+MBSE-P silver^b	66.9±0.2	31.0
+201 domain gold sent.	72.3±0.2	68.0
+403 domain gold sent.	74.3±0.2	70.0
+5K domain gold sent.	79.8±0.2	80.0
+MBSE-G silv.^{b}+201 gold	75.8±0.3	70.0
+MBSE-G silv.^{b}+5K gold	**81.3**±0.2	**81.0**
QALD-9-AMR Evaluations		
Ours		
Struct-BART (AMR3.0)	87.2	84.0
+MBSE-G silver^1	88.0	**88.0**
+MBSE-G silver^q	89.5±0.1	85.0
+MBSE-P silver^q	89.3±0.2	87.0
+200 domain gold sent.	88.5±0.5	84.0
+408 domain gold sent.	89.8±0.1	86.0
+MBSE-G silv.^{q}+200 gold	90.0±0.3	87.0
+MBSE-G silv.^{q}+408 gold	**90.1**±0.1	87.0

Table 6: Smatch scores on BioAMR and QALD-9 test sets with varying sizes of human annotated (gold) domain sentences and silver data. MBSE-G (greedy-select) and MBSE-P (Graphene-Smatch respectively). MBSE distillations are all with Ensemble-4 (APT + Structured-BART + SPRING$_1$ + SPRING$_2$).

Regarding QALD-9-AMR data, MBSE distillation on silver^q, i.e. SQuAD-Q(uestion) sentences,

Models	BioAMR	Q9AMR
Structured-BART	8.7%	0.7%
+MBSE silver	3.7%	0.7%
+200 domain gold sents	0.6%	0.7%

Table 7: Named entity (NE) type out-of-vocabulary ratio w.r.t the target vocabulary of various models. BioAMR and QALD-9 test sets include 1691 and 150 occurrences of named entities, respectively.

is almost as effective as 408 domain gold sentences (89.8) for both graphene-Smatch (89.3) and greedy-select (89.5) algorithms. Combining 408 domain gold sentences with MBSE greedy-select silver^q adds less than 1 Smatch point to 90.1.

Since MBSE distillation on silver^b lags behind the performance of 201 human annotated AMR for BioAMR, mostly due to low NER scores, we further analyze the target vocabulary coverage of named entity (NE) types occurring in the test sets. The analysis is shown in Table 7. NE types are equally well covered in all models for Q9AMR (QALD-9-AMR). 0.7% out-of-vocabulary (OOV) ratio is caused by a typo in human annotation of the test set, i.e. *country* misspelled as *contrty*. For BioAMR, however, NE type OOV ratio of MBSE silver model is 3.7%, e.g. *protein-segment*, *macro-molecular-complex*, substantially higher than 0.6% of the model trained with 201 domain gold sentences. When the NE type is OOV, there is no chance for the system to produce the missing NE type, let alone predicting it correctly, underscoring the challenges posed by domain specific concepts unavailable elsewhere.

6 Related Work

There have been numerous works applying ensemble/knowledge distillation (Hinton et al., 2015) to machine translation (Kim and Rush, 2016; Freitag et al., 2017; Nguyen et al., 2020; Wang et al., 2020, 2021), dependency parsing (Kuncoro et al., 2016) and question answering (Mun et al., 2018; Ze et al., 2020; You et al., 2021; Chen et al., 2012). Regarding ensembling AMR graphs, Barzdins and Gosko (2016) propose choosing the AMR with highest average sentence Smatch to all other AMRs. Hoang et al. (2021) proposed a more complex technique capable of building new AMRs by exploiting Smatch's hill climbing algorithm. Our work brings together ensemble distillation and Smatch-based ensembling and shows that it can provide substantial gains over the standard self-training.

Damonte and Cohen (2018) show that it may be possible to use the original AMR annotations devised for English as representations of equivalent sentences in other languages. Damonte and Cohen (2018); Sheth et al. (2021) propose annotation projection of English AMR graphs to target languages to train cross-lingual parsers, using word alignments. Blloshmi et al. (2020) show that one may not need alignment-based parsers for cross-lingual AMR, and model concept identification as a *seq2seq* problem. Procopio et al. (2021) reframe semantic parsing as multilingual machine translation (MNMT) and propose a seq2seq architecture fine-tuned on pretrained-mBART with an MNMT objective. Cai et al. (2021b) propose to use bilingual input to enable a model to predict more accurate AMR concepts. Xu et al. (2021) propose a cross-lingual pretraining approach via multitask learning for AMR parsing. Cai et al. (2021a) propose to use noisy knowledge distillation for multilingual AMR parsing. We introduce Structured-mBART and attain new SoTA in Chinese, German, Italian and Spanish cross-lingual parsing by applying MBSE distillation and AMR-to-text.

We subsume domain adaptation under data augmentation with MBSE distillation, where the only difference between the two lies in the properties of the unlabeled data. The unlabeled data is drawn from the target domain for the purpose of domain adaptation rather than those similar to the source training data for data augmentation in general.

7 Conclusion

We proposed a technique called Maximum Bayes Smatch Ensemble (MBSE) distillation, which brings together Smatch-based model ensembling Barzdins and Gosko (2016); Hoang et al. (2021) and ensemble distillation Hinton et al. (2015) of heterogeneous parsers, to significantly improve AMR parsing. The technique generalizes well across various tasks and is highly effective, leading to a new single system SoTA in English and cross-lingual AMR parsing and achieving the performance comparable to human annotated training data in domain adaptation of QALD-9-AMR corpus. Remaining technical challenges include tokenization and alignment of an input token corresponding to more than one concept for AMR parsing and identification of unknown named entities and their types for domain adaptation.

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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.

Guntis Barzdins and Didzis Gosko. 2016. Riga at semeval-2016 task 8: Impact of smatch extensions and character-level neural machine translation on amr parsing accuracy. In *Proceedings of SemEval-2016*, pages 1143–1147.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *AAAI Technical Track on Speech and Natural Language Processing I, Vol. 35 No. 14*, pages 12564–12573.

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.

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

Deng Cai, Xin Li, Jackie Chun-Sing Ho, Lidong Bing, and Wai Lam. 2021a. Multilingual amr parsing with noisy knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2778–2789.

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.

Cen Chen, Chengyu Wang, Minghui Qiu, Dehong Gao, Linbo Jin, and Wang Li. 2012. Cross-domain knowledge distillation for retrieval-based question answering systems. In *Proceedings of the World Wide Web Conference 2021*.

K. B. Cohen, K. Verspoor, K. Fort, C. Funk, M. Bada, M. Palmer, and L. E. Hunter. 2017. The colorado richly annotated full text (craft) corpus: multi-model annotation in the biomedical domain. *Handbook of Linguistic Annotation*, pages 1379–1394.

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.

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

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. In *arXiv:1702.01802*.

Vaibhava Goel and William J Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Computer Speech & Language*, 14(2):115–135.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam M. Nguyen, Dzung T. Phan, Vanessa López, and Ramon Fernandez Astudillo. 2021. Ensembling graph prediction for amr parsing. *arXiv preprint arXiv:2110.09131*.

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.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Junichi Tsujii. 2011. Overview of bionlp shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 66–75.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. Technical report, JOHNS HOPKINS UNIV BALTIMORE MD CENTER FOR LANGUAGE AND SPEECH PROCESSING (CLSP).

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Distilling an ensemble of greedy dependency parsers into one mst parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1744–1753.

Young-Suk Lee, Ramon Fernandez Astudillo, Tahira Naseem, Revanth Gangi Reddy, Radu Florian, and Salim Roukos. 2020. Pushing the limits of amr parsing with self-learning. In *Findings of the EMNLP2020*, pages 3208–3214.

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 *Proceesings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. In *arXiv:2001.08210*.

Chunchuan Lyu, Shay B Cohen, and Ivan Titov. 2020. A differentiable relaxation of graph segmentation and alignment for amr parsing. *arXiv preprint arXiv:2010.12676*.

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.

Jonghwan Mun, Kimin Lee, Jinwoo Shin, and Bohyung Han. 2018. Learning to specialize with knowledge distillation for visual question answering. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*.

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.

Xuan-Phi Nguyen, Shafiq Joty, Wu Kui, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, pages 48–53.

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 (ACL)*, pages 311–318.

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.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the Third Workshop on Natural Language Processing for Conversational AI*, pages 211–227.

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

Janaki Sheth, Young-Suk Lee, Ramon 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*, pages 394–404.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Sarah Uhrig, Yoalli Rezepka Garća, 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 (IWPT 2021)*, pages 58–64.

Ricardo Usbeck, Ria Hari Gusmita, Muhamad Saleem, and Axel-Cyrille Ngonga Ngomo. 2018. 9th challenge on question answering over linked data (qald-9). In *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4, Vol-2241*, pages 58–64.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for 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*, pages 6456–6466.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020. Transductive ensemble learning for neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 6291–6298.

Andrew Gordon Wilson and Pavel Izmailov. 2020. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*.

Qingrong Xia, Zhenghua Li, Rui Wang, and Min Zhang. 2021. Stacked amr parsing with silver data. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4729–4738.

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*, pages 2501–2511.

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*, pages 896–907.

Chenyu You, Nuo Chen, and Yuexian Zou. 2021. Knowledge distillation for improved accuracy in spoken question answering. In *ICASSP 2021*.

Yang Ze, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2020. Model compression with two-stage multi-teacher knowledge distillation for web question answering system. In *The Thirteenth ACM International Conference on Web Search and Data Mining*.

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.

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 (EMNLPIJC-NLP)*, pages 3784–3796.

Jiawei Zhou, Tahira Naseem, Ramon Fernandez Astudillo, and Radu Florian. 2021a. 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.

Jiawei Zhou, Tahira Naseem, Ramon Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021b. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based amr parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6290.

A Greedy-Select Ensemble Algorithm

Algorithm 1: Greedy-Select MBSE Algorithm and Corpus Selection

```
Input: AMR$_1$...AMR$_n$ parses from $n$ AMR parsing models, where $n \geq 3$
Optionally Require: Smatch score threshold = $\theta$
Output: One-best AMR parse
 1: Let bestAMR = $NULL$
 2: for $\forall_{i,j}$ in $1 \leq i, j \leq n$ and $i \neq j$ do
 3:     Compute sentence Smatch score $smatch$(AMR$_i$, AMR$_j$), total $n(n-1)/2$ scores.
 4:     Pick the highest $smatch$(AMR$_i$, AMR$_j$).
 5:     for Each AMR$_a$, where $a = i$ or $a = j$ do
 6:         Pick the highest $smatch$(AMR$_a$, AMR$_b$)
 7:         if $a = i$ then
 8:             bestAMR = AMR$_i$
 9:         else
10:             bestAMR = AMR$_j$
11:         end if
12:         if $smatch$(AMR$_a$,AMR$_b$) $< \theta$ then
13:             bestAMR = $NULL$
14:             {no AMR to be used from this sentence}
15:         end if
16:     end for
17: end for
18: return bestAMR
```

We start with n parses from n heterogeneous parsing models, where the minimum number of parses is 3. For each input sentence, we compute sentence-level Smatch scores between any two parses across all n parses, for a total of $n(n-1)/2$ Smatch scores (lines 2-3). Subsequently, we pick the two parses AMR$_i$ and AMR$_j$ with the highest Smatch score, where AMR$_i$ denotes the AMR parse from the system i (line 4) For each of the two parses, AMR$_i$ and AMR$_j$, we choose the parse with the higher Smatch score against the rest of the parses as the best parse (lines 5-11). When the scores are tied, we select the first parse output (equivalent to a random choice of fixed seed).

We incorporate an optional parse selection criterion into Algorithm 1, indicated as **Optionally Require** and specified in lines 12-15. The bestAMR for input sentence is selected for data augmentation if the Smatch score $smatch$(AMR$_a$, AMR$_b$) is greater than or equal to the pre-specified value θ.

Models	AMR2.0	AMR3.0
Structured-BART	34,156	33,200
SPRING$_1$	25,129	29,407
SPRING$_2$	17,866	17,830
APT	10,235	6,949
Total	87,386	87,386

Table 8: Distribution of individual model parses from MBSE greedy-select distillation with silver^1 dataset in Table 1

Model	Param	AMR2.0	AMR3.0
sep-voc	src voc size	50,265	50,265
	tgt voc size	42,344	42,784
	# param	493,011,968	493,913,088
joint-voc	joint voc size	57,912	58,673
	# param	414,121,984	414,901,248

Table 9: Vocabulary and parameter sizes of Structured-BART with MBSE distillation on silver^{1+2+3} dataset from Table 1

B Model Structures and Parameter Size

Pre-trained BART and mBART share the same model configurations except for the vocabulary size. There are 12 encoder/decoder layers, 16 heads per layer, 1024 model dimension and 4096 feed forward network (FFN) size. BART includes ~50K and mBART, ~250K task vocabulary.

When using separate vocabulary (sep-voc), Structured-BART and Structured-mBART use the same vocabulary as BART and mBART, respectively, for the source. For the target, they create embedding vectors for action symbols and the target vocabulary size vary according to the training data. When using joint vocabulary (joint-voc), Structured-BART shares the same vocabulary between the source and the target, a combination of BART vocabulary and the additional embedding vectors for some action symbols.

Vocabulary and parameter sizes for Structured-BART and Structured-mBART trained with MBSE distillation are shown in Table 9 and Table 10, respectively.

C Implementation Details

Our models are implemented with FAIRSEQ toolkit (Ott et al., 2019), trained and tested on a single NVIDIA Tesla A100/V100 GPU with 40-80GB memory. We use fp16 mixed precision training and all models are trained on 1 GPU.

For all English AMR parsing models with silver data, we use the Adam optimizer with $\beta_1 = 0.9$

Languages	voc size	# param
DE (German)	34,689	681,894,912
ES (Spanish)	34,881	682,288,128
IT (Italian)	33,681	679,830,528
ZH (Chinese)	59,473	732,652,544

Table 10: Target vocabulary (sep-voc) and parameter sizes of Structured-mBART with MBSE distillation on silver^{1+2+3} dataset from Table 1. Source vocabulary size is 250,027 across all languages.

Lgs.	vocab	base model	ens. model
EN	joint-voc	60min	60min
DE	sep-voc	23min	42min
ES	sep-voc	24min	44min
IT	sep-voc	22min	40min
ZH	sep-voc	30min	60min

Table 11: Inference time for AMR2.0 test set. Base models are trained on AMR2.0 treebank only and ens. models are trained on AMR2.0 treebank plus silver^{1+2+3}.

and $\beta_2 = 0.98$. Batch size is set to 1024 maximum number of tokens with gradient accumulation over 8 steps. Learning rate schedule is the same as Vaswani et al. (2017) with 4000 warm-up steps and $1e{-7}$ warm-up initial learning rate and the maximum learning rate $1e{-4}$. Dropout rate is 0.2 and label smoothing rate is 0.01. These hyper parameters are fixed and not tuned for different models and datasets. All models are trained for 10 epochs and the best 5 checkpoints are selected based on the development set Smatch from greedy decoding. Model parameters are averaged over the top 3 and top 5 models. The model that produces the highest development set score, after beam search decoding with beam size = 1, 5 and 10, is selected as the final model. Training with MBSE greedy-select silver^{1+2+3} takes 48-72 hours, and all other models with less silver data take less time to train.

For cross-lingual AMR parsing, maximum learning rate is always set to $3e{-5}$. Baseline models trained only on AMR2.0 corpus are trained up to 80 epochs whereas models with silver^1 (and AMR-to-text) is trained up to 30 epochs and models with silver^{1+2+3}, up to 15 epochs. Model parameters are updated after gradient is accumulated for 8192 tokens. Dropout rate, label smoothing rate and model selection criteria are the same as the English parsers. Training baseline models takes about 10 hours. Training with silver^1 takes about 24 hours. Training with silver^{1+2+3} takes about 96-120 hours.

In order to reduce the vocabulary size, which subsequently reduces the model parameter size and memory requirement, we prune out singleton target vocabulary for training with silver data.

Inference time for AMR2.0 benchmark test set is shown in Table 11, where beam size=10 and batch size=64 for all languages. EN is decoded on NVIDIA Tesla A100 and all other languages, on NVIDIA Tesla V100.