SBERT studies Meaning Representations: Decomposing Sentence Embeddings into Explainable Semantic Features

Juri Opitz
Dept. of Computational Linguistics
Heidelberg University
69120 Heidelberg
opitz.sci@gmail.com

Anette Frank
Dept. of Computational Linguistics
Heidelberg University
69120 Heidelberg
frank@cl.uni-heidelberg.de

Abstract

Models based on large-pretrained language models, such as S(entence)BERT, provide effective and efficient sentence embeddings that show high correlation to human similarity ratings, but lack interpretability. On the other hand, graph metrics for graph-based meaning representations (e.g., Abstract Meaning Representation, AMR) can make explicit the semantic aspects in which two sentences are similar. However, such metrics tend to be slow, rely on parsers, and do not reach state-of-the-art performance when rating sentence similarity.

In this work, we aim at the best of both worlds, by learning to induce Semantically Structured Sentence BERT embeddings (S^{3}BERT). Our S^{3}BERT embeddings are composed of explainable sub-embeddings that emphasize various semantic sentence features (e.g., semantic roles, negation, or quantification). We show how to i) learn a decomposition of the sentence embeddings into semantic features, through approximation of a suite of interpretable AMR graph metrics, and how to ii) preserve the overall power of the neural embeddings by controlling the decomposition learning process with a second objective that enforces consistency with the similarity ratings of an SBERT teacher model. In our experimental studies, we show that our approach offers interpretability – while fully preserving the effectiveness and efficiency of the neural sentence embeddings.

1 Introduction

Abstract Meaning Representation (AMR) represents the meaning of a sentence as a directed, rooted and acyclic graph (Banarescu et al., 2013). It shows events and entities referred to in a sentence, their semantic roles and key semantic relations such as *cause, time, purpose, instrument, negation*.

The explicit representation of meaning in AMR has motivated research into AMR metrics that measure meaning similarity of the underlying sentences. E.g., AMR metrics are used for semantics-focused NLG evaluation (Opitz and Frank, 2021; Manning and Schneider, 2021; Zeidler et al., 2022), a semantic search engine (Bonial et al., 2020), comparison of cross-lingual AMR (Uhrig et al., 2021; Wein et al., 2022), and argument similarity (Opitz et al., 2021b). Moreover, fine-grained AMR metrics can assess meaning similarity of semantic sub-aspects that AMR explicitly captures, e.g., semantic roles or negation (Damonte et al., 2017).

However, when measuring similarity rating performance against human ratings in the typical zero-shot setting on tasks like STS (Baudiš et al., 2016a) or SICK (Marelli et al., 2014), the (untrained) AMR metrics tend to lag behind large models such as SBERT (Reimers and Gurevych, 2019) that computes sentence embeddings with a Siamese BERT transformer model (Devlin et al., 2019).

Notably, SBERT alleviates the need for end-to-end similarity inference on each sentence pair. Instead, it infers the embedding of each sentence individually, and calculates similarity with simple vector algebra, which greatly reduces clustering and search time. AMR metrics, by contrast, tend to be slower, are often NP-hard (Cai and Knight, 2013) and rely on a parser.

Hence, we find complementarity in these two approaches of rating sentence similarity: AMR metrics offer high explainability – but tend to be slow and need improvement to compete in benchmarking. By contrast, neural embeddings show strong empirical performance and efficiency – but lack explainability.

Aiming at the best of these worlds, we propose to leverage multi-aspect AMR metrics as a means to teach a pre-trained SBERT model on how to structure its sentence embedding space such that it explicitly captures specific abstract aspects of meaning similarity, in terms of semantic roles, negation, quantification, etc. This has to be undertaken with care, to prevent catastrophic forgetting (Goodfellow et al., 2013; Hayes et al., 2020), which could

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 625–638
November 20–23, 2022. ©2022 Association for Computational Linguistics

negatively impact SBERT's empirical performance and the overall effectiveness of its embeddings.

Our contributions:

1. To increase the explainability of sentence embeddings, we propose a method that performs *Semantic Decomposition* in the SBERT sentence embedding space, to yield S^{3}BERT (Semantically Structured SBERT) embeddings. S^{3}BERT sub-embeddings express key semantic sentence features that reflect AMR metric measurements taken on the sentences' underlying meaning representations.
2. To prevent catastrophic forgetting, we include a consistency objective that controls the decomposition learning process and projects important semantic information not captured by AMR to a residual sub-embedding.
3. Our experiments and analyses in zero-shot sentence and argument similarity tasks show that S^{3}BERT embeddings are more explainable than SBERT embeddings while fully preserving SBERT's efficiency and accuracy.
4. Code and data are publicly released: https://github.com/flipz357/S3BERT

2 Related work

SBERT and friends: High efficacy at the cost of lower interpretability Since its introduction by Reimers and Gurevych (2019), S(entence)BERT has become a popular method for computing sentence similarity (Thakur et al., 2020; Reimers and Gurevych, 2020; Wang and Kuo, 2020; Seo et al., 2022). This is due to two key properties: SBERT shows strong results on similarity benchmark tasks and it is highly efficient. E.g., it allows rapid sentence clustering since the BERT backbone is called independently for each sentence, alleviating the need for pair-wise model inferences.

However, SBERT provides little explainability. While different linguistic indicators have been identified for or within BERT (Jawahar et al., 2019; Lepori and McCoy, 2020; Warstadt et al., 2019; Puccetti et al., 2021), this insight by itself does not provide us with any rationale for high (or low) sentence similarity in specific cases, and so, to achieve *local* explainability (Danilevsky et al., 2020), we would have to, at least, analyze attention weights (Clark et al., 2019; Wiegreffe and Pinter, 2019) or gradients (Selvaraju et al., 2017; Sanyal and Ren, 2021; Bastings and Filippova, 2020) of regions associated with linguistic properties. But even then, it can be unclear how exactly to interpret the results (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Wang et al., 2020; Ferrando and Costa-jussà, 2021). In a different direction, Kaster et al. (2021) aim to explain BERTscore (Zhang et al., 2020) predictions with a regressor. But unlike other explanation methods, this approach is detached from the underlying BERT model and may suffer from indirection effects. Instead, we target local self-explainability (Danilevsky et al., 2020) by structuring SBERT's sentence embedding space into subspaces that emphasize explicit facets of meaning. Parts of this idea are inspired from Rothe and Schütze (2016), who compose four semantic spaces of *word vectors*, using a lexical resource. Without such a resource, and targeting sentence embeddings, we aim to leverage and structure semantic knowledge already present in the model, while injecting new knowledge that we obtain from metrics grounded in a multi-faceted theory of meaning, namely AMR.

AMR metrics: the cost of interpretability AMR graphs (Banarescu et al., 2013) explicate aspects of meaning, such as entities, events, coreference, or negation. Metrics defined over AMRs therefore show specific aspects in which two sentences are similar or different, which makes them attractive for tasks going beyond parser evaluation, such as NLG evaluation (Opitz and Frank, 2021; Manning and Schneider, 2021), semantic search (Bonial et al., 2020), explainable argument similarity rating (Opitz et al., 2021b), or investigation of cross-lingual divergences (Uhrig et al., 2021; Wein et al., 2022). While classical AMR metrics assess semantic similarity structurally via binary matches of triples (Cai and Knight, 2013), recent metrics target larger contexts and graded similarity scoring (Opitz et al., 2020, 2021a), e.g., to match a subgraph *cat :mod young* against a node *kitten*.

But this high degree of explainability comes at a price: AMR metrics tend to be slow since they i) compute costly graph alignments (Cai and Knight, 2013) and/or ii) require AMR parsers (Opitz et al., 2022) that are typically slow due to auto-regressive inference of large LMs (Raffel et al., 2019; Lewis et al., 2019). iii) They are untrained, and thus tend to lag behind SBERT-based metrics in empirical settings (Opitz et al., 2021a). We aim to overcome these weaknesses by making sentence embeddings capable of expressing AMR metrics while preserving the full power of neural sentence embeddings.

Sentence and argument similarity Several works and resources aim to capture human sentence similarity ratings. E.g., SICK (Marelli et al., 2014) rates *semantic relatedness* and STS (Baudiš et al., 2016a) *semantic similarity*, on 5-point Likert scales. *Relatedness* and *Similarity* have been argued to be very similar notions, albeit not the exact same (Budanitsky and Hirst, 2006; Kolb, 2009).[1]

An emergent branch of sentence similarity is the similarity of natural language arguments (Reimers et al., 2019; Opitz et al., 2021b; Behrendt and Harmeling, 2021), which finds broad application scenarios, e.g., in argument search engines (Maturana, 1988; Wachsmuth et al., 2017; Ajjour et al., 2019; Lenz et al., 2020; Slonim et al., 2021).

While much research has been devoted to improving the accuracy of similarity rating systems, little attention has been paid to uncovering the features that (in the eyes of a human) make two sentences similar or dissimilar (Zeidler et al., 2022). In our work, we propose a method that can potentially help uncover such features, while provably preserving strong rating accuracy.

3 From SBERT to S^{3}BERT: Structuring embedding space with AMR

Preliminary I: SBERT sentence embeddings and similarity Let SB be a function that maps an input sentence s to a vector $e \in \mathbb{R}^d$. Given two sentence vectors $e = SB(s)$ and $e' = SB(s')$, we can compute, e.g., the cosine similarity of sentences:

$$sim(e, e') = \frac{e^T e'}{|e||e'|}. \tag{1}$$

Preliminary II: AMR and AMR metrics An AMR $a \in A$ represents the meaning of a sentence in a directed acyclic graph. The AMR graph makes key aspects of meaning explicit, e.g., semantic roles or negation. Hence, given a pair of AMR graphs $\langle a, a' \rangle \in A \times A$, an AMR metric can measure *overall* graph similarity, or similarity with respect to *specific aspects*. We denote such a metric as

$$m^k : A \times A \rightarrow [0, 1], \tag{2}$$

where k indicates a particular semantic aspect, in view of which the graphs' similarity is assessed, e.g. negation. The AMR metrics we will apply in our work will be described in more detail in §4.

3.1 Partitioning sentence embeddings into meaningful semantic AMR aspects

Problem statement We aim to shape SBERT sentence embeddings in such a way that different sub-embeddings represent specific meaning aspects. This process of *sentence embedding decomposition* is illustrated in Fig. 1 (right): SBERT produces two embeddings e and e' that consist of sub-embeddings $F_1...F_K, R$ and $F'_1...F'_K, R'$. E.g., F_k may express negation features, while F_z expresses semantic role features of a sentence. The residual R offers space to model sentence features not covered by the pre-defined set of semantic features.

Having established such decompositions, we can compute, e.g., sentence similarity with respect to semantic roles ($k = SRL$) by choosing subspaces $F_{SRL} \subset e = SB(s)$ and $F'_{SRL} \subset e' = SB(s')$, and calculating $sim(F_{SRL}, F'_{SRL})$ on the subspaces. This is indicated as ▶◀ in Fig. 1.

Assigning embedding dimensions to features For convenience, let $i : \{1...K\} \rightarrow [0, d] \times [0, d]$ denote an AMR aspect-embedding assignment function where d is the dimension of the (full) sentence embedding. This allows us to map any semantic category to a range of specific sentence embedding indices. E.g., a h-dimensional embedding for SRL sentence features for a sentence s can be accessed via $SB(s)_{i(SRL)}$, where $v_{(start,end)}$ yields all dimensions from $start$ to end of a vector v. Since we aim at a non-overlap decomposition, we ensure that $i(k) \cap i(k') \neq \emptyset \iff k = k'$.

3.2 Learning to partition the semantic space

We presume that SBERT already contains some semantic features in some embedding dimensions. Hence, we want to achieve an arrangement of the embedding space according to our pre-defined partitioning, but also give it the chance to instill new knowledge about AMR semantics.

In addition, to preserve SBERT's high accuracy, we aim to control the decomposition process in a way that lets us route internal semantic knowledge *not* captured by AMR to the residual embedding. To this end, we propose a two-fold objective: *Score decomposition* and *Score consistency*.

Composing S^{3}BERT score from AMR metrics We build an AMR metric target $\mathbf{M}$ as shown in Fig. 1 (left). Two AMRs, constructed from two sentences, are assessed with AMR metrics in K semantic aspects (Eq. 2) yielding $\mathbf{M} \in \mathcal{M} = \mathbb{R}^K$. Ad-

[1] Only the highest rating on the SICK and STS Likert scales mean the exact same: two sentences are equivalent in meaning.

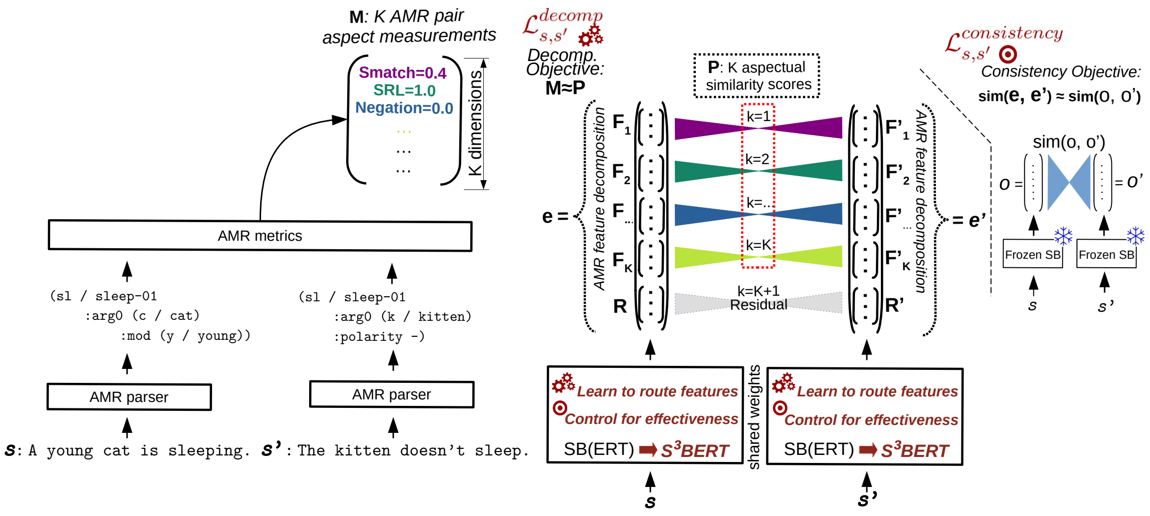
M: K AMR pair aspect measurements
Smatch=0.4
SRL=1.0
Negation=0.0
...
K dimensions
AMR metrics
(sl / sleep-01 :arg0 (c / cat) :mod (y / young))
(sl / sleep-01 :arg0 (k / kitten) :polarity -)
AMR parser
AMR parser
s: A young cat is sleeping.
s': The kitten doesn't sleep.
$\mathcal{L}^{decomp}_{s,s'}$
Decomp. Objective: M≈P
P: K aspectual similarity scores
k=1
k=2
k=...
k=K
k=K+1 Residual
e =
AMR feature decomposition
F1
F2
F...
FK
R
F'1
F'2
F'...
F'K
R'
= e'
$\mathcal{L}^{consistency}_{s,s'}$
Consistency Objective: sim(e, e') ≈ sim(o, o')
sim(o, o')
o =
= o'
Frozen SB
Frozen SB
s
s'
Learn to route features
Control for effectiveness
SB(ERT) ➡ S3BERT
shared weights
Learn to route features
Control for effectiveness
SB(ERT) ➡ S3BERT
s
s'

Figure 1: Overview of approach. ⚙ The decomposition objective structures the sentence embedding space into AMR sentence features (F$_1$...F$_K$): The process is guided by AMR metric approximation, through which S^{3}BERT learns to disentangle and route the features. ⊙ The consistency objective is aimed at preventing catastrophic forgetting: To preserve the overall effectiveness of the neural sentence embeddings, it controls the decomposition learning process and helps modeling the residual (R).

ditionally, let **P** be S^{3}BERT's AMR metric predictions, i.e., $\mathbf{P} = [sim(F_1, F_1'), ..., sim(F_K, F_K')]$.

For a training instance $(s, s', \mathbf{M})$, we calculate the following decomposition loss:

$$\mathcal{L}^{decomp}_{s,s'} = \qquad (3)$$

$$\frac{1}{K}\sum_{k=1}^{K}\Big[\mathbf{M}_k - \beta^k \underbrace{sim(SB(s)_{i(k)}, SB(s')_{i(k)})}_{\mathbf{P}_k}\Big]^2,$$

with β^k a learnable scalar for easier projection onto a specific AMR metric's scale. The objective is also outlined as **P**≈**M** in Fig. 1.

Note that AMR graphs and metrics are only needed for training, not for inference.

3.3 Preventing catastrophic forgetting

When training S^{3}BERT only with the *decomposition objective* (Eq. 3), there is a great risk it will unlearn important information, since it is unrealistic to expect that sentence similarity can be *fully* composed from the K aspects measured by AMR metrics. It is also known that AMR metrics lag behind SBERT models in similarity rating accuracy. Hence, we control the decomposition learning process to include a *residual* sub-embedding, to rescue important parts of semantic information not captured by AMR and AMR metrics. To this end, we propose a *consistency objective*.

Given a frozen SBERT ($SB^{❄}$), and a training example (s, s'):

$$\mathcal{L}^{consistency}_{s,s'} = \Big(sim(SB^{❄}(s), SB^{❄}(s')) - sim(SB(s), SB(s'))\Big)^2.$$

I.e., the control is established by imposing that S^{3}BERT's overall similarity ratings be in accordance with a frozen SBERT's original ratings, but otherwise leaving freedom for the choice of structure in S^{3}BERT's embedding space. Given independence of pairwise-targets, we can compute the loss efficiently on b^2 examples in batches of size b.

3.4 Global objective

We finally combine the *consistency objective* and the *decomposition objective*. The cumulative loss for a batch $B = \{(S_i, S_i', \mathcal{M}_i)\}_{i=1}^{b}$ is

$$\mathbf{L} = \frac{\alpha}{b}\sum_{i=1}^{b}\mathcal{L}^{decomp}_{S_i,S_i'} + \frac{1}{b^2}\sum_{i=1}^{b}\sum_{j=1}^{b}\mathcal{L}^{consistency}_{S_i,S_j'}, \qquad (4)$$

where α weighs the two parts (we use $\alpha = 1$).

4 AMR metrics and data construction

In Section 3, Eq. 2, we formally described an AMR metric. Now we consider the concrete metric instances we will use for S^{3}BERT decomposition. We distinguish *general* metrics that assess global AMR graph similarity, and *aspectual* metrics that aim at assessing AMR similarity with respect to specific semantic categories, e.g., semantic roles.

4.1 Global AMR similarity

SMATCH assesses the structural overlap of two semantic AMR graphs. It computes a best fitting combinatorial alignment between AMR variable nodes and returns a triple overlap score.

WLKERNEL and WWLKERNEL Opitz et al. (2021a) apply the structural Weisfeiler-Leman kernel (Weisfeiler and Leman, 1968; Shervashidze et al., 2011) aiming at more contextualized AMR graph matches. The method extracts sub-graph statistics from the input graphs that describe different levels of node contextualizations. To assess a modulated similarity of AMR graphs, Opitz et al. (2021a) adapt the Wasserstein Weisfeiler-Leman metric (Togninalli et al., 2019), which compares the graphs in a joint latent space using the (permutation-invariant) Wasserstein distance.

4.2 Aspectual AMR similarity

FINESMATCH: Fine-grained SMATCH Damonte et al. (2017) create fine-grained SMATCH-based metrics to analyze AMR similarity w.r.t. interesting semantic categories. We use **Frames**: graph similarity with regard to PropBank predicates. **Named entity**: graph similarity based on named entity substructures (*person, city, ...*). **Negation**: graph similarity based on expressions of negation. **Concepts**: graph similarity based on node labels only. **Coreference**: graph similarity focused on co-referent structures. **SRL**: graph similarity considering predicate substructures. Finally, **Unlabeled**: not considering semantic edge labels.[2]

Additionally, we observe that AMR contains information about quantifiers and define **quantSim**, which measures the (normalized) overlap of quantifier structure of two AMRs. Although AMR lacks modeling of quantifier scope (Bos, 2016), estimating the overlap of quantificational structure can give indications of semantic sentence similarity.

Graph statistics In addition, we introduce graph metrics that target other aspects modeled by AMR: **MaxIndegreeSim, maxOutDegreeSim** and **maxDegreeSim**. From each graph in a pair of AMRs, we extract the node that is best connected (either outdegree, indegree, or indegree+outdegree). We compare these nodes with cosine similarity using GloVe embeddings (Pennington et al., 2014). The motivation for this is that two Meaning Representations that share the same focus are more likely to be similar (Lambrecht, 1996). Similarly, **rootSim** compares the similarity of AMR roots, motivated by Cai and Lam (2019), who speculate that more important concepts are closer to the root.

4.3 Data setup

For the decomposition objective we need training instances of paired sentences with AMR metric scores attached. We proceed as follows:

1) We collect 1,500,000 sentence pairs from data sets that contain similar sentences.[3] 2) We parse these sentences with a good off-the-shelf AMR parser.[4] 3) For each training sentence pair we create a positive (a, a^+) and a negative (a, a^-) datum, where the negative pair is formed by replacing AMR a^+ with an AMR sampled from a random pair. Thereby we show S^{3}BERT both AMR metric outputs computed from similar AMRs, and unrelated AMRs (that may still share some abstract semantic features). 4) We execute our AMR metrics (c.f. §4.1 & §4.2) over all pairs from step 3). Step 4) took approx. 3 days, since AMR metrics tend to have high computational complexity.

For experimentation, we cut off a development and testing set with 2,500 positive pairs each.[5]

5 Evaluation Study

Our two objectives aim at creating S^{3}BERT embeddings by partitioning SBERT's output space into features that capture different semantic AMR aspects, while controlling the decomposition process such that we prevent any forgetting of knowledge and preserve the power of the neural embeddings.

Hence, two key questions need to be addressed:

1.) Will S^{3}BERT partition its sentence embedding space into interpretable semantic aspects?

2.) If so, what is the price? Does our consistency objective succeed in controlling the decomposition process such that it retains SBERT's extraneous knowledge of sentence semantics?

[2] We follow Opitz (2020) and set metric values to 1.00 (as opposed to 0.00) in cases where neither of the graphs contains structures of the given aspect (e.g., named entities are absent from both graphs), since the graphs can then be considered to (vacuously) agree in the given aspect.

[3] AllNLI, CoCo, flickr captions, quora duplicate questions.

[4] `https://github.com/bjascob/amrlib` The parser is based on a fine-tuned T5 (Raffel et al., 2019) language model and reports more than 80 Smatch points on AMR3. On a GPU Ti 1080 the parsing took approx. 3 weeks.

[5] Using only similar sentence pairs for validation increases the AMR metric prediction difficulty and provides a useful lower bound for correlation.

Basic setup We use a standard SBERT model[6] with 11 layers and allow tuning of the last two layers. The sentence embedding dimension is $d = 384$, the sub-embedding dimension is set to $h = 16$ for all 15 aspects of AMR, which implies that the dimension of the residual is $384 - (15 \times 16) = 144$. More details on the model architecture and the training hyper-parameters can be found in Appendix A.1. In all result tables, † indicates statistically significant improvement over the runner-up (Student t-test, $p < 0.05$, five random runs)

5.1 S^{3}BERT space partitioning

Our goal is to make SBERT embeddings more interpretable, by partitioning the sentence embedding space into multiple semantically meaningful sub-embeddings. We now aim to answer research question **1)** whether these sub-embeddings relate to the AMR metric aspects they were trained to predict.

Data setup We use the 2,500 testing sentence pairs we had split from our generated data. For each semantic aspect, we calculate cosine similarities of the corresponding sub-embeddings. We then calculate the Spearmanr correlation of these predictions vs. the ground truth AMR metric similarities.

Baseline setup We consider three baselines. Same as S^{3}BERT, all baselines are based on standard SBERT model.[6]

SB-full (no partitioning): We use the complete embedding, which means that we predict the same value for all AMR aspects. This baseline is bound to provide strong correlations with most metrics[7], but obviously lacks the interpretability we are aiming for. We therefore instantiate two more baselines that can be directly compared, since they partition the space according to semantic aspects.

SB-rand (partitioning): We assign 16 embedding dimensions randomly to every semantic aspect.

SB-ILP (partitioning): We use an integer linear program to assign the semantic aspects to different SBERT dimensions. We create a bi-partite weighted graph with node sets (V_{SB}, V_{SEM}) with SBERT dimensions (V_{SB}), and the targeted semantic aspects (V_{SEM}). Then, we introduce weighted edges $(i, j) \in V_{SB} \times V_{SEM}$, where a weight $\omega(i, j)$ is the Spearmanr correlation of SBERT values in dimension i vs. the metric scores for aspect j across all (development) data instances. We solve (5–7).

		partitioning models		
aspect	SB-full	SB-rand	SB-ILP	S^{3}BERT
SMATCH	64.6	57.1	57.9	***68.2***†
WLKERNEL	*76.7*†	63.5	64.2	**74.6**
WWLKERNEL	*75.1*	62.0	63.8	**74.4**
Frames	46.0	40.8	45.2	***66.4***†
Unlabeled	58.4	52.3	54.7	***65.1***†
Named Ent.	-14.4	-1.1	-0.3	***51.1***†
Negation	-2.00	-0.0	3.4	***33.0***†
Concepts	*76.7*†	64.5	72.3	**74.0**
Coreference	23.2	10.3	13.6	***43.3***†
SRL	48.3	40.8	44.9	***60.8***†
maxIndegreeSim	27.0	23.6	24.0	***32.5***†
maxOutDegreeSim	22.3	17.5	19.4	***42.5***†
maxDegreeSim	22.3	18.0	19.7	***30.0***†
rootSim	25.5	21.7	25.1	***43.1***†
quantSim	11.5	10.0	11.8	***74.6***†

Table 1: Spearmanr x 100 of AMR aspects. *Italics*: overall best. **bold**: best partitioning approach. underlined: improvement by more than 20 Spearmanr points.

$$\max \sum_{(i,j) \in V_{SB} \times V_{SEM}} \omega(i, j) \cdot x_{ij} \tag{5}$$

$$s.t. \sum_{j} x_{ij} \leq 1 \; \forall i \in V_{SB} \tag{6}$$

$$\sum_{i} x_{ij} \geq 1 \; \forall j \in V_{SEM} \tag{7}$$

The binary decision variables $x_{ij} \in \{0, 1\}$ indicate whether an SBERT dimension is part of a specific sub-embedding. The first constraint decomposes SBERT embeddings into non-overlapping parts, one for each aspect. The second constraint ensures that each semantic aspect is modeled.

Results are displayed in Table 1. First, we see that the global AMR metrics WLKERNEL and WWLKERNEL are best modeled with the cosine distance computed on full SBERT embeddings (unpartitioned, Table 1) and we can't model them as well with a sub-embedding. This seems intuitive: the power of a low-dimensional sub-embedding is too low to express the complexity of the two Weisfeiler graph metrics that aim at capturing broader AMR sub-structures. However, the structural SMATCH, which does not match structures beyond triples, can be better modeled in a sub-embedding (+3.8 vs. SB-full). Nonetheless, compared to the best partitioning baseline (SB-ILP), our approach provides substantial improvements (Spearmanr points, WLKERNEL +10.4, WWLKERNEL +10.6).

Therefore, it is more interesting to study the fine-grained semantic aspects measured by our aspectual AMR metrics. We find that there are three

[6] Pre-trained `All-MiniLM-L12-v2` from the sentence transformers library.

[7] Since AMR metrics correlate with human sentence similarity (Opitz et al., 2021a), and so does SBERT.

AMR features that are very poorly modeled with global SBERT embeddings: *named entities*, *negation*, *quantification*. They also cannot be extracted with the SB-ILP baseline. By contrast, S^{3}BERT clearly improves over these baselines. E.g., *negation* modeling improves from a negative correlation to a significant positive correlation of 33.0 Spearmanr. *Quantifier similarity* increases from 11.8 Spearmanr to 74.6.

5.2 Correlation with human judgements

Relating to research question **2)** on whether we can effectively prevent SBERT from forgetting prior knowledge when teaching it to predict AMR metrics, we test how well our approach compares to human ratings of sentence similarity in the typical zero shot setting. As our main goal is to increase the interpretability of SBERT predictions, we consider S^{3}BERT achieving SBERT's original performance on this task a satisfying objective.

5.2.1 Sentence semantic similarity

Test data We use sentence semantic similarity data with human ratings. The STS (STSb) benchmark (Baudiš et al., 2016b) assesses semantic similarity and SICK (Marelli et al., 2014) relatedness.[8]

Evaluation metric We again use Spearmanr. To assess *efficiency*, we display the approximate time for a metric to process 1,000 pairs. We also want to assess the *explainability* of the methods, which can be complicated (Danilevsky et al., 2020). To keep it as simple as possible, we assign ★★ when a metric is fully transparent and the score can be traced in the meaning space via graph alignment (SMATCH, WWLKERNEL), and ★ if there is a dedicated mechanism of explanation (e.g., via a linguistically decomposable score, as in S^{3}BERT).

Baselines As baselines we use: 1. SBERT and 2. our S^{3}BERT from which we ablate a) the decomposition objective (S^{3}BERT$^{\cancel{\text{dec}}}$) or b) the consistency objective (S^{3}BERT$^{\cancel{\text{cons.}}}$). Assessing S^{3}BERT$^{\cancel{\text{cons.}}}$ is key, since it shows the performance when we only focus on learning AMR features – a significantly reduced score would prove the importance of counter-balancing decomposition with our consistency objective. For reference, we also include results from a simplistic baseline (word overlap) and the AMR metrics computed from the AMR graphs of sentences as in Opitz et al. (2021a).

system	speed (1k pairs)	xplain	STSb	SICK
bag-of-words	0s	-	43.2	53.3
bag-of-nodes	31m (p) + 0.0s (i)	-	60.4	61.6
SMATCH	31m (p) + 49s (i)	★★	57.2	59.1
WLKERNEL	31m (p) + 1s (i)	-	63.9	61.4
WWLKERNEL	31m (p) + 5s (i)	★★	62.5	64.7
SBERT	1s (i)	-	83.1	78.9
S^{3}BERT	1s (i)	★	**83.7**†	**79.1**
S^{3}BERT$^{\cancel{\text{dec}}}$	1s (i)	-	83.0	78.9
S^{3}BERT$^{\cancel{\text{cons.}}}$	1s (i)	★	51.7	58.1

Table 2: Results on STSb and SICK using Spearmanr x 100; Speed measurements of parser (p) and metric inference (i), units are minutes (m) and seconds (s).

		3-Likert	binary classif. F1 scores		
system	xplain	Spea's r	Macro	Sim	¬ Sim.
RE19	-	-	65.4	52.3	78.5
BH21	-	34.8	-	-	-
OP21	★★	-	68.6	60.4	77.0
SBERT	-	54.2	71.7	63.8	79.6
S^{3}BERT	★	**56.4**†	**72.9**†	**65.7**†	**80.1**†
S^{3}BERT$^{\cancel{\text{cons.}}}$	★	28.2	55.6	53.7	57.4

Table 3: Results on argument similarity prediction.

Results are shown in Table 2. Interestingly, while one main goal was to prevent a performance drop, S^{3}BERT tends to outperform all baselines, including SBERT (significant improvement for STSb).

It is important to note that catastrophic forgetting indeed occurs if learning is not controlled by the consistency objective. In this case, the performance drops by about 20-30 points (S^{3}BERT$^{\cancel{\text{cons.}}}$ in Table 2). We conclude that our consistency objective effectively prevented any loss of embedding power.

5.2.2 Argument similarity

Testing data Besides the STS and SICK benchmarks we use the challenging UKPA(spect) data (Reimers et al., 2019) with high-quality similarity ratings of natural language arguments from 28 controversial topics such as, e.g., *GMO* or *Fracking*.

Evaluation metric Argument pairs in UKPA have one of four labels: *dissimilar, unrelated, somewhat similar* and *highly similar*. Originally, the task was evaluated as a binary classification task (Reimers et al., 2019), by mapping the *similar* and *highly similar* labels to 1, and the other two labels to zero. A similarity metric's scores are then mapped to binary decisions via a simple threshold-search script. To conform with this work, we also evaluate using this setup. But to account for

[8] We min-max normalize the Likert-scale ratings of both datasets to the range between 0 and 1.

the fine-grained labels, we also use a second metric based on (Spearmanr) correlation, following Behrendt and Harmeling (2021) who propose a 3-Likert scale that maps *dissimilar* and *unrelated* to 0, *somewhat similar* to 0.5, *highly similar* to 1.0.

Baselines Table 3 shows the results of the best systems reported for i) a BERT-based approach (Reimers et al., 2019) (RE19), ii) the AMR-based SMATCH-variant approach of Opitz et al. (2021b), and iii) Behrendt and Harmeling (2021) (BH21), who pre-train BERT on other argumentation datasets for 3-Likert style rating.

Results S^{3}BERT significantly outperforms all baselines, including SBERT, in the classification setting, and in the correlation evaluation setting. When assessing interpretability, OP21 offers ★★ because it is based on SMATCH and the score can be *fully* traced. However, it is less efficient, due to the cost of executing AMR metrics and parser, and lags behind in accuracy. Again, we can conclude that our approach offers a valuable balance between interpretability and performance. Finally, this experiment further corroborates that controlling the decomposition learning process is paramount: without consistency objective, the accuracy is almost halved (S^{3}BERT$^{\cancel{cons.}}$ in Table 3).

5.3 Ablation and parametrization experiments

Upper-bounds for AMR metric approximation While not the main objective of our work, the approximation of computationally expensive AMR metrics can be considered an interesting task on its own. We hence explore two AMR metric approximation upper-bounds: i) *S^{3}BERT$^{\cancel{cons.}}$*: Naturally, the consistency objective is orthogonal to the AMR metric approximation objective and by ablating the consistency objective, we can obtain an upper-bound for the prediction of AMR metric scores. *ii) S^{3}BERT$^{\cancel{cons.}}$+parser*: At the cost of making our approach much less efficient, we train S^{3}BERT$^{\cancel{cons.}}$ directly on (linearized) AMR graph strings instead of their underlying sentences, which allows us to infer metric scores directly from AMR graphs.

The results of these setups are given in Table 6 in Appendix A.3. We see that both modifications can yield, to some extent, better AMR metric approximation accuracy, across all tested aspects. However, considering our second key goal of preserving the overall power of sentence embeddings, it is important to note that these improvements come at great cost, because if we do not control the decomposition process with our consistency objective, the similarity rating effectivity of the neural embeddings deteriorates (see S^{3}BERT$^{\cancel{cons.}}$ in Table 2 for sentence similarity and Table 3 for argument similarity). On top of this, S^{3}BERT$^{\cancel{cons.}}$+parser will also lose much *efficiency*.[9]

Effect of parser quality For creating AMRs, we used a strong parser that yields high SMATCH scores on AMR benchmarks. To investigate the effect of using another parser, we re-ran our first experiment (decomposition) with metrics computed from parses of the older JAMR (Flanigan et al., 2014) parser, that achieves more than 20 points lower SMATCH on AMR benchmarks. We observe moderately(+1-3 correlation points) better results across all categories with the more recent parser. This implies that there is potential room for further improvement of our method by using an even more accurate parser, but judging from the marginally lower score of JAMR, the gain may be small.

Size of training data We observe that the AMR metric approximation accuracy profits from growing size of the training data (see Appendix A.2).

6 Data analyses with S^{3}BERT

6.1 Studying S^{3}BERT predictions

We find many interesting cases where S^{3}BERT is able to explain its similarity scores.[10] For example, both S^{3}BERT and SBERT assign a high similarity score (0.70–0.73) to *two cats are looking at a window* vs. *a white cat looking out of a window*, while the human similarity rating is just above average (.52). Here, a low similarity rating of -0.15 in S^{3}BERT's **quantifier feature** provides a (possible) rationale for the much lower human score, due to a strong contrast in quantifier meaning (*two* vs. *a*).

When confronted with **negation**, both SBERT and S^{3}BERT assign moderately high scores to *The man likes cheese* vs. *the man doesn't like cheese*. But S^{3}BERT can explain this: its high *concept* similarity score increases the overall rating, while a (very) low similarity score for *negation* (-0.30) regulates the rating downwards. We also see differences in how negation of a matrix verb affects the S^{3}BERT negation feature – compared with negation applied to a sub-ordinate sentence. *Three boys in karate costumes [aren't | are] fighting* results in

[9] Due to slow AMR parsing (c.f. Table 2).

[10] See more examples in Table 7, Appendix A.4.

		aspectual semantic feature											global AMR feature				
FEASIM	data	Conc.	Frame	NE	Neg.	Coref	SRL	IDgr	ODgr	Dgr	$\sqrt{Sim}$	quant	Sma.	Unlab.	WLK	W^{2}LK	Resid.
vs. HUM	STSb	**73.8**$_{(1)}$	68.7	60.4	53.6	65.6	70.8$_{(2)}$	66.8	64.8	69.9$_{(3)}$	67.2	51.6	72.7	68.1	**75.1**	72.8	**83.3**
vs. SIM	STSb	**88.3**$_{(1)}$	81.5	75.6	61.9	80.0	84.4$_{(2)}$	81.2$_{(3)}$	78.7	81.2$_{(3)}$	77.5	60.1	86.1	83.4	**88.9**	86.4	**99.3**
vs. HUM	UKP	51.3	**61.3**$_{(1)}$	26.9	52.1$_{(3)}$	42.9	43.7	33.6	57.1$_{(2)}$	42.0	45.4	-4.2	30.3	**37.8**	10.9	25.2	26.1
vs. SIM	UKP	**98.3**$_{(1)}$	86.7	85.0	93.3$_{(2)}$	91.7	90.0	90.0	91.7$_{(3)}$	85.0	86.7	63.3	**91.7**	86.7	81.7	86.7	**96.7**

Table 4: Similarity investigation with S^{3}BERT feature analysis. **bold**/(n): best from a feature group (rank 1–3).

lower negation agreement (Negation feature similarity: -0.31) compared to negation applying to the predicate of a sub-ordinate sentence, as in *A child is walking down the street and a jeep [is not | is] pulling up* (Negation feature similarity: -0.22).

Coreference can also explain key differences in meaning: *The cat scratches a cat* and *The cat scratches itself* are highly rated in all aspects (0.78–0.8 overall similarity) – except for coreference, with similarity of only 0.41, signaling a key difference reflected in coreference structures.

Comparing the **foci of sentences** can also provide explanatory information. E.g., the human score for *a man is smoking* and *a baby is sucking on a pacifier* is zero, indicating complete dissimilarity. But S^{3}BERT and SBERT assign scores that indicate moderate similarity. S^{3}BERT's features may explain this, in that the sentences' foci (root sim) are somewhat related (0.4, *smoking* vs. *sucking*).

6.2 Studying predictors of human scores

What features can predict *human similarity scores* and how may the assessment of argument similarity as opposed to sentence similarity differ from each other? In search for answers to these questions, we perform a quantitative analysis of S^{3}BERT's fine-grained features. We proceed as follows: Let *SIM* be S^{3}BERT's similarity ratings for a pairwise data set, and *HUM* be the corresponding human ratings. Now, let *FEASIM* be the fine-grained S^{3}BERT feature similarities for a feature *FEA* (e.g., SRL aspect). Then we compute, for each *FEA*, *Spearmanr(FEASIM, SIM)* and *Spearmanr(FEASIM, HUM)*, both on STS and argumentation benchmarks. In other words, we analyze predictive capacity of features for a) system vs. b) human similarity in c) different domains/tasks.

Analysis results are shown in Table 4. Interestingly, for *human argument similarity*, the residual has much lower predictive power (26.1), suggesting that human argument similarity notions differ significantly from sentence similarity. Indeed, another key difference can be found in the importance of quantification similarity, which is marginal (-4.2) for argumentation, but not for STS (51.6). We speculate that users judging argument similarity tend to generalize over quantifier differences, being more focused on general statements and concepts, as opposed to, e.g., numerical precision. Notably, human argument similarity is markedly well predicted by **Frames** – this feature alone achieves state-of-the-art results, indicating a marked importance of predicate frames for argument similarity.

Of course, although the analysis may give some interesting indications about similarity as perceived by humans (and SBERT), it has to be taken with a grain of salt, one reason being, e.g., that the shown statistics are influenced by AMR metric prediction accuracy, which varies across aspects (c.f. Table 1). Our study also indicates that neither sentence nor argument similarity can be fully explained by any feature. We hypothesize that we may need to go beyond what SBERT and (current) AMR metrics can measure, e.g., by incorporating background knowledge. Our method may offer a way to inject such background knowledge into sentence embeddings, via distillation of dedicated metrics.

7 Conclusion

We propose a method for decomposing neural sentence embedding spaces into different sub-spaces, with the goal of obtaining sentence similarity ratings that are *accurate, efficient* and *explainable*. The sub-spaces express facets of meaning as captured by AMR and AMR metrics, such as *Negation* or *Semantic Roles*. The *decomposition objective* partitions the semantic space via targeted synthesis of AMR metrics. The effectiveness of neural sentence embeddings is preserved by a *consistency objective* that controls the decomposition process and routes global semantic information not expressed by AMR into a *residual embedding*. The S^{3}BERT embeddings are more explainable and are on par, or even outperform, SBERT's accuracy. Our approach allows straightforward extension to customized metrics of meaning similarity.

Acknowledgements

We are grateful to our anonymous reviewers for their comments that have helped to significantly improve our paper. We also thank Julius Steen and Philipp Wiesenbach for helpful feedback on ideas. The work has been partially funded by the DFG through the project ACCEPT as part of the Priority Program "Robust Argumentation Machines" (SPP 1999).

References

Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. 2019. Data acquisition for argument search: The args. me corpus. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 48–59. Springer.

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.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Petr Baudiš, Jan Pichl, Tomáš Vyskočil, and Jan Šedivỳ. 2016a. Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*.

Petr Baudiš, Silvestr Stanko, and Jan Šedivý. 2016b. Joint learning of sentence embeddings for relevance and entailment. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 8–17, Berlin, Germany. Association for Computational Linguistics.

Maike Behrendt and Stefan Harmeling. 2021. Argue-BERT: How to improve BERT embeddings for measuring the similarity of arguments. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 28–36, Düsseldorf, Germany. KONVENS 2021 Organizers.

Claire Bonial, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare Voss. 2020. 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.

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

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational linguistics*, 32(1):13–47.

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.

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.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. 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.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

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.

Javier Ferrando and Marta R. Costa-jussà. 2021. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 434–443, Punta Cana, Dominican Republic. 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.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. 2020. Remind your neural network to prevent catastrophic forgetting. In *Computer Vision – ECCV 2020*, pages 466–483, Cham. Springer International Publishing.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Marvin Kaster, Wei Zhao, and Steffen Eger. 2021. Global explainability of BERT-based evaluation metrics by disentangling along linguistic factors. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8912–8925, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Kolb. 2009. Experiments on the difference between semantic similarity and relatedness. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 81–88, Odense, Denmark. Northern European Association for Language Technology (NEALT).

Knud Lambrecht. 1996. *Information structure and sentence form: Topic, focus, and the mental representations of discourse referents*, volume 71. Cambridge university press.

Mirko Lenz, Premtim Sahitaj, Sean Kallenberg, Christopher Coors, Lorik Dumani, Ralf Schenkel, and Ralph Bergmann. 2020. Towards an argument mining pipeline transforming texts to argument graphs. *Computational Models of Argument: Proceedings of COMMA 2020*, 326:263.

Michael Lepori and R. Thomas McCoy. 2020. Picking BERT's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Emma Manning and Nathan Schneider. 2021. Referenceless parsing-based evaluation of AMR-to-English generation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 114–122, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Humberto R Maturana. 1988. Reality: The search for objectivity or the quest for a compelling argument. *The Irish journal of psychology*, 9(1):25–82.

Juri Opitz. 2020. AMR quality rating with a lightweight CNN. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 235–247, Suzhou, China. Association for Computational Linguistics.

Juri Opitz, Angel Daza, and Anette Frank. 2021a. Weisfeiler-Leman in the Bamboo: Novel AMR Graph Metrics and a Benchmark for AMR Graph Similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.

Juri Opitz, Philipp Heinisch, Philipp Wiesenbach, Philipp Cimiano, and Anette Frank. 2021b. Explainable unsupervised argument similarity rating with Abstract Meaning Representation and conclusion generation. In *Proceedings of the 8th Workshop on Argument Mining*, pages 24–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juri Opitz, Philipp Meier, and Anette Frank. 2022. SMARAGD: Synthesized sMatch for Accurate and Rapid AMR Graph Distance. *arXiv preprint arXiv:2203.13226*.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. Amr similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Giovanni Puccetti, Alessio Miaschi, and Felice Dell'Orletta. 2021. How do BERT embeddings organize linguistic knowledge? In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 48–57, Online. Association for Computational Linguistics.

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. *CoRR*, abs/1910.10683.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Sascha Rothe and Hinrich Schütze. 2016. Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–517, Berlin, Germany. Association for Computational Linguistics.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Jaejin Seo, Sangwon Lee, Ling Liu, and Wonik Choi. 2022. Ta-sbert: Token attention sentence-bert for improving sentence representation. *IEEE Access*, 10:39119–39128.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9).

Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, et al. 2021. An autonomous debating system. *Nature*, 591(7850):379–384.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*.

Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. 2019. Wasserstein weisfeiler-lehman graph kernels. In *Advances in Neural Information Processing Systems*, volume 32, pages 6436–6446. Curran Associates, Inc.

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.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59.

Bin Wang and C.-C. Jay Kuo. 2020. Sbert-wk: A sentence embedding method by dissecting bert-based word models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2146–2157.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of NLP models is manipulable. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages

247–258, Online. Association for Computational Linguistics.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. 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 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Shira Wein, Wai Ching Leung, Yifu Mu, and Nathan Schneider. 2022. Effect of source language on AMR structure. In *Proceedings of The 16th Linguistic Annotation Workshop (LAW)*, Marseille, France. European Language Resources Association (ELRA).

Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Laura Zeidler, Juri Opitz, and Anette Frank. 2022. A dynamic, interpreted CheckList for meaning-oriented NLG metric evaluation – through the lens of semantic similarity rating. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 157–172, Seattle, Washington. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

A.1 Hyper-parameters and training

Batch size is set to 64, the learning rate (after 100 warm-up steps) is set to 0.00001. We train for 8 epochs, evaluating every 1000 steps. Afterwards we select the model from the evaluation step where we achieve minimum development loss.

A.2 Scaling training data size

See Table 5.

A.3 AMR metric approximation upper-bounds

See Table 6.

aspect	amount of training data			
	rand (0k)	50k	300k	1500k
SMATCH	57.1	59.4	60.2	68.2
WLKERNEL	63.5	64.1	70.2	74.6
WWLKERNEL	62.0	65.8	67.0	74.4
Frames	40.8	44.2	53.6	66.4
Unlabeled	52.3	53.6	54.1	65.1
Named Ent.	-1.1	11.4	31.8	51.1
Negation	-0.0	17.8	29.0	33.0
Concepts	76.7	69.6	71.2	74.0
Coreference	23.2	23.9	25.2	43.3
SRL	48.3	49.4	50.0	60.8
maxIndegreeSim	27.0	26.7	26.4	32.5
maxOutDegreeSim	22.3	22.4	23.1	42.5
maxDegreeSim	22.3	22.1	22.5	30.0
rootSim	25.5	26.4	28.9	43.1
quantSim	11.5	47.1	65.4	74.6

Table 5: AMR prediction performance w.r.t. different training data sizes.

aspect	S^3BERT	S^3BERT$^{\cancel{cons.}}$	S^3BERT$^{\cancel{cons.}}$+parser
SMATCH	68.2	77.0	80.3
WLKERNEL	74.6	79.3	78.9
WWLKERNEL	74.4	81.5	82.3
Frames	66.4	79.6	80.3
Unlabeled	65.1	75.5	78.0
Named Ent.	51.1	58.0	61.9
Negation	33.0	34.5	35.5
Concepts	74.0	78.5	76.4
Coreference	43.3	57.4	72.1
SRL	60.8	74.3	83.0
maxIndegreeSim	32.5	37.3	37.5
maxOutDegreeSim	42.5	59.9	65.4
maxDegreeSim	30.0	40.6	42.7
rootSim	43.1	57.4	81.2
quantSim	74.6	75.7	76.1

Table 6: AMR metric approximation upper-bounds. *S^3BERT$^{\cancel{cons.}}$*: S^3BERT without consistency objective (trades sentence similarity rating performance for better AMR approximation). *S^3BERT$^{\cancel{cons.}}$+parser*: S^3BERT without consistency objective and inference on linearized AMR graphs (trades sentence similarity rating performance *and* efficiency for better AMR approximation).

index	sentence pairs	humSim	SBERT	S^{3}BERT	notable feature similarities
1	two cats are looking at a window a white cat looking out of a window	0.52	0.70	0.72	concepts: 0.87↑↑; quant: -0.15↓↓
2	three men posing in a tent three men eating in a kitchen	0.24	0.39	0.42	quant:0.99↑↑; Frames: -0.02↓↓, Unlabeled: 0.6 ↑
3	rocky and apollo creed are running down the beach the men are jogging on the beach	0.6	0.33	0.32	maxDegSim: 0.4↑, NamedEnt: -0.72↓↓
4	a man is smoking a baby is sucking on a pacifier	0.0	0.06	0.06	rootSim↑↑: 0.4
5	a dog prepares to herd three sheep with horns a dog and sheep run together	0.44	0.63	0.65	SRL: 0.56↓; Frames: 0.45↓, Concepts: 0.85↑
6	The cat scratches itself The cat scratches another cat	na	0.81	0.78	Concepts: 0.9 ↓; Negation 0.56↓; Coref: 0.41↓↓
7	The man likes cheese The man doesn't like cheese	na	0.80	0.77	Concepts: 0.90 ↑; Negation: -0.3 ↓↓
8	Recruits are talking to an officer An officer is talking to the recruits	0.68	0.97	0.98	SRL: 0.96 ↓; Negation: 0.90 ↓; Unlabeled: 0.99 ↑
9	A dog is teasing a monkey at the zoo A monkey is teasing a dog at the zoo	0.63	0.99	0.99	SRL: 0.96 ↓; Negation: 0.97 ↓; maxDegr: 1.0 ↑
10	Three boys in karate costumes aren't fighting Three boys in karate costumes are fighting	0.58	0.86	0.86	Concepts: 0.92↑; Negation: -0.31↓↓
11	A child is walking down the street and a jeep is pulling up A child is walking down the street and a jeep is not pulling up	0.63	0.95	0.92	Concepts: 0.95↑; Negation: -0.22↓↓

Table 7: Prediction Examples from STSb and SICK, or own construction (human rating: na).

A.4 Prediction examples

See Table 7.