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

Juri Opitz
Dept. of Computational Linguistics
Heidelberg University
69120 Heidelberg
opitz@cl.uni-heidelberg.de

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

Abstract

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

In this paper, we aim at the best of these two worlds, by creating similarity metrics that are highly effective, while also providing an interpretable rationale for their rating. Our approach works in two steps: We first select AMR graph metrics that measure meaning similarity of sentences with respect to key semantic facets, such as, i.a., semantic roles, negation, or quantification. Second, we employ these metrics to induce Semantically Structured Sentence BERT embeddings (S$^3$BERT), which are composed of different meaning aspects captured in different sub-spaces. In our experimental studies, we show that our approach offers a valuable balance between performance and interpretability.

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. For instance, AMR metrics are used for semantics-focused NLG evaluation (Opitz and Frank, 2021; Manning and Schneider, 2021), a COVID-19 search engine (Bonial et al., 2020), comparison of cross-lingual AMR (Uhrig et al., 2021), and argument similarity (Opitz et al., 2021b).

However, when measuring sentence similarity against human ratings in empirical data sets such as STS (Baudiš et al., 2016a; Cer et al., 2017) 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 model (Devlin et al., 2019).

Notably, SBERT alleviates the need for end-to-end 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 (Nagarajan and Sviridenko, 2009; Cai and Knight, 2013) and rely on a parser.

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

Aiming at the best of these worlds, we propose to distill AMR metrics into a pre-trained SBERT model such that it arranges its sentence embedding dimensions to explicitly capture specific aspects of meaning similarity, such as semantic roles, negation, or quantification. This has to be undertaken with care, to prevent catastrophic forgetting (Goodfellow et al., 2013; Hayes et al., 2020), which could negatively impact SBERT’s empirical performance.

Our contributions:

1. To increase the explainability of SBERT, we propose a method that performs explicit Meaning Decomposition in the SBERT sentence embedding space, to yield S$^3$BERT
(Semantically Structured SBERT). The decomposed sub-embeddings hold semantic aspects as measured by AMR metrics on the underlying sentences’ meaning representations.

2. To prevent catastrophic forgetting, we include a consistency objective that projects important semantic information not captured by AMR or AMR metrics to a residual sub-embedding.

3. Our experiments show that S3BERT embeddings are more explainable than SBERT embeddings while preserving SBERT’s efficiency and accuracy in two tasks: sentence and argument similarity.

4. Our code and data will be publicly released.

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: It shows strong results on similarity benchmark tasks such as STS (Baudiš et al., 2016a; Cer et al., 2017) or SICK (Marelli et al., 2014) and it is highly efficient. E.g., SBERT can perform rapid sentence clustering, since the BERT backbone is called independently for each sentence, alleviating the need for quadratic 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, and so, to achieve local explainability (Danilevsky et al., 2020), we would have to, e.g., analyze attention weights (Clark et al., 2019; Wiegrefe and Pinter, 2019) or gradients (Selvaraju et al., 2017; Sanyal and Ren, 2021; Bastings and Filippova, 2020) of regions that can be associated with a specific linguistic property. Yet, even then, it can be unclear how exactly to interpret the results (Jain and Wallace, 2019; Wiegrefe and Pinter, 2019; Wang et al., 2020; Ferrando and Costajussà, 2021). In a different direction, Kaster et al. (2021) aim to explain BERTscore (Zhang et al., 2020) predictions with linguistic features in a linear 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 capture explicit facets of meaning. Parts of this idea are inspired from Rothe and Schütte (2016), who compose four meaning spaces of word vectors, using a lexical resource. Without such a resource, and targeting sentence embeddings, we want to leverage and structure semantic knowledge already in the model, while also providing the chance to add new knowledge, with our approach being grounded in a single theory of meaning, AMR.

AMR metrics: the cost of interpretability AMR (Banarescu et al., 2013) represents sentence meaning in a graph, explicating entities, events, and other semantic aspects such as coreference or negation. Accuracy metrics defined over AMR graphs therefore uncover specific aspects in which two sentences are similar or different. This makes them attractive for tasks beyond parser evaluation: AMR metrics have been proposed for NLG evaluation (Opitz and Frank, 2021; Manning and Schneider, 2021), in a COVID19 search engine (Bonial et al., 2020), explainable argument similarity rating (Opitz et al., 2021b), or for inspecting 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 include larger contexts and graded similarity scoring (Opitz et al., 2020, 2021a), e.g., to assess how similar a subgraph cat:mod young is to a node kitten.

But this high degree of interpretability 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, 2020; Opitz et al., 2022) that are slow due to auto-regressive inference of long sequences (Raffel et al., 2019; Lewis et al., 2019). iii) They are untrained, and thus tend to lag behind SBERT-based metrics when reproducing human similarity ratings (Opitz et al., 2021a). We aim to overcome these weaknesses by injecting AMR metrics into SBERT.

Sentence and argument similarity Several works and resources aim to capture human sentence similarity ratings. E.g., SICK (Marelli et al., 2014) measures semantic relatedness and STS (Baudiš et al., 2016a) measures semantic similarity, both using a 5-point Likert scale. Relatedness and Simi-
larity have been argued to be very similar notions, albeit not the exact same (Budanitsky and Hirst, 2006; Kolb, 2009). Only the highest rating on the SICK and STS Likert scales mean the exact same: two sentences are equivalent in meaning.

An interesting of computational sentence similarity is the similarity rating of natural language arguments (Reimers et al., 2019; Opitz et al., 2021b; Behrendt and Harmeling, 2021) – which finds a broad application scenario in argument search engines and argument analysis (Rissland et al., 1993; 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 uncover 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 uncover such features, while preserving strong rating accuracy.

3 From SBERT to S\textsuperscript{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 the cosine similarity of sentences:

\[
sim(e, e') = \frac{e^T e'}{|e||e'|}.
\]

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 an AMR metric as

\[
m^k : A \times A \to [0, 1],
\]

where \( k \) indicates a particular semantic aspect or category, in view of which the the graphs’ similarity should be assessed. In our case, we consider general graph similarity and similarity with respect to aspects such as semantic roles (SRL), negation, etc. The AMR metrics that we apply will be described in more detail in Section 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 Figure 1 (right-hand side): SBERT produces two sentence embeddings \( e, e' \) that consist of sub-embeddings \( F_1...F_K, R \), respectively, \( F'_1...F'_K, R' \). Here, \( F_k \) may express semantic roles, while \( R \) expresses semantic role features of a sentence. The residual \( R \) offers space to model sentence features that are 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 using Eq. 1 to calculate \( \sim(F_{SRL}, F'_{SRL}) \) on the subspaces. This is indicated as \( \nrightarrow \) in Fig. 1.

Assigning embedding dimensions to features For convenience, let \( i : \{1...K\} \to [0, d] \times [0, d] \) 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 \( v_{(\textit{start}, \textit{end})} \) yields all dimensions from \( \textit{start} \) to \( \textit{end} \) 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 has already captured some semantic features in specific dimensions. In a hypothetical edge-case where it models all given AMR metrics perfectly, the task for S\textsuperscript{3}BERT would be to arrange its output dimensions such that they fit our assignment. Hence, we want to allow SBERT to arrange its output embeddings according to our pre-defined partitioning, while giving it the chance to instill new knowledge about AMR and AMR metrics. In addition, we allow it to route internal semantic knowledge not captured by AMR to the residual embedding, to preserve SBERT’s high accuracy. To this end, we propose a two-fold objective: Score decomposition and Score consistency.

Composing S\textsuperscript{3}BERT score from AMR metrics The creation of our AMR metric targets is outlined.
in Figure 1 (left). Two AMRs, constructed from two sentences, are assessed with AMR metrics for K semantic aspects (Eq. 2) to produce a score vector of target scores containing m benign aspects.

For a training instance (s, s', M), we calculate the following decomposition loss:

\[
\mathcal{L}_{\text{decomp}}^{(s, s')} = \frac{1}{K} \sum_{k=1}^{K} \left[ M_k - \beta_k \frac{\sum_{i \in M_k} \text{sim}(SB(s)_{i(k)}, SB(s')_{i(k)})}{\sum_{i \in M_k} 1} \right]^2.
\]

with \(\beta_k\) a learnable scalar parameter that scales SBERT similarities for given subfeatures \(k\). The objective is given as \(P=M\) in Fig. 1: \(P\) indicates the vector of target scores containing \(m^k(a, a') \forall k \in \{1...K\}\); \(P\) indicates the \(K\)-dimensional vector of SBERT’s predictions computed with cosine similarity (Eq. 1) over the semantic sub-embeddings: \([\text{sim}(F_1, F'_1)...\text{sim}(F_K, F'_K)]\). Note that AMR graphs and metrics are only needed for training, not for inference.

### 3.3 Preventing forgetting with a permutation invariant consistency loss

When training SBERT to minimize only the decomposition objective (Eq. 3), it could forget important information, since it is unrealistic to expect that sentence similarity can be fully composed into the \(K\) aspects measured by AMR metrics. Indeed, the residual sub-embedding might retain important parts of such information. To give SBERT some guidance on how to model the residual, we employ a consistency objective. Given a frozen SBERT model \(SB^\star\), and a training example \((s, s')\),

\[
\mathcal{L}_{\text{consistency}}^{(s, s')} = \left( \frac{\sum_{i \in M_k} \text{sim}(SB^\star(s), SB^\star(s'))}{\sum_{i \in M_k} 1} - \frac{\sum_{i \in M_k} \text{sim}(SB(s), SB(s'))}{\sum_{i \in M_k} 1} \right)^2.
\]

Since this objective is independent of a pairwise-target, we can compute it fast on \(b^2\) examples contained in a batch of size \(b\).

### 3.4 Global objective

Our global objective combines the consistency objective and the decomposition objective. The cumulative loss for a batch \(B = \{(s_i, s'_i, M_i)\}_{i=1}^b\) is

\[
L = \frac{\alpha}{b} \sum_{i=1}^{b} \mathcal{L}_{\text{decomp}}^{(s_i, s'_i)} + \frac{1}{b^2} \sum_{i=1}^{b} \sum_{j=1}^{b} \mathcal{L}_{\text{consistency}}^{(s_i, s'_j)},
\]

where \(\alpha\) can weigh 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** The SMATCH metric assesses the structural overlap of two AMR graphs in two steps. It computes an alignment between variable nodes of the AMRs and based on the obtained alignment, it assesses triple matches to compute an F1 score.
We compare these nodes with cosine similarity using the Wasserstein distance between the two graphs. AMRs, we extract the node that is best connected which measures the (normalized) overlap of quantificational structures of the given aspect (e.g., named entities ... 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). The graphs can then be considered to agree in the given aspect.

Additionally, we observe that AMR contains information about quantifiers and define quantSim, which measures the (normalized) overlap of quantifiers of two AMR graphs. 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 pairwise sentences and assigned AMR metric scores that rate the similarity of the corresponding AMR graphs. We proceed as follows:

1) We collect 1,500,000 sentence pairs from data sets that contain similar sentences. 2) We parse these sentences with a good off-the-shelf AMR parser that is based on a fine-tuned T5 (Raffel et al., 2019) language model and reports more than 80 Smatch points on AMR3. 3) For each sentence pair we create a positive $(a, a^+)$ and a negative $(a, a^-)$ datum, where the positive datum is the original pair and the negative pair is formed by replacing AMR $a^+$ with an AMR sampled from another similar pair. The idea is that we want to show SBERT both AMR metric outputs computed from similar AMRs, as well as unrelated AMRs (that may still share some abstract semantic features). 4) We execute the 14 AMR metrics described in Sections 4.1 and 4.2 over all data obtained from step 3).

Step 4) took us approx. 3 days, since AMR metrics may still have high computational complexity.

Now we have large training data consisting of sentence pairs and AMR metric scores as they emerge from comparing the AMR sentence graphs. For experimentation, we cut off a development and testing set consisting of 2,500 data instances each.

5 Evaluation Study
Two important questions need to be addressed:

1. Will S3BERT partition its sentence embedding space into interpretable AMR aspects? 2. If so, at what cost? I.e., will S3BERT forget relevant sentence semantics when learning the embedding decomposition?

General model setup Our two objectives aim at partitioning SBERT’s output space such that a prediction can be composed from different semantic
aspects, and at the same time at preventing the forgetting of knowledge that is not modeled by AMR (metrics). We use a standard SBERT model\(^4\) with 11 layers, where we only tune 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. We use \(†\) to indicate statistical significance of a result (Student t-test, \(p < 0.05\), five random runs).

### 5.1 SBERT 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 Spearman correlation of the predicted against the ground truth AMR metric similarities.

**Baseline setup** We consider three baselines. All baselines are based on the standard SBERT model.

**SB-full (no partitioning)**: We use all embedding dimensions for computing cosine distance, which means that we predict the same value for all AMR aspects. This baseline is likely to provide strong correlations with most metrics\(^5\), but it obviously lacks the interpretability that we are aiming for by not partitioning the embedding space. We therefore instantiate two more baselines that can be directly compared, since they do 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 programming optimization approach to assign the semantic similarity aspects to different SBERT dimensions. We create a bi-partite weighted graph with node sets \((V_{SB}, V_{SEM})\) that identifies SBERT sentence dimensions on one side, and the targeted semantic similarity aspects on the other. Then, we introduce weighted edges \((i, j) \in V_{SB} \times V_{SEM}\).

---

\(\)\(^4\)All-MiniLM-L12-v2

\(\)\(^5\)Since AMR metrics correlate with human sentence similarity (Opitz et al., 2021a), and so does SBERT.

---

| aspect       | unpartitioned | partitioning models |
|--------------|--------------|---------------------|
|              | SB-full      | SB-rand | SB-ILP | SB-ours |
| S\(\)\(^3\)MATCH | 64.6\(^†\) | 57.1  | 57.9  | 60.2   |
| W\(\)LKERNEL  | 76.7\(^†\)  | 63.5  | 64.2  | 70.2   |
| W\(\)LKERNEL  | 75.1\(^†\)  | 62.0  | 63.8  | 67.0   |
| Frames       | 46.0         | 40.8  | 45.2  | 53.6\(^†\) |
| Unlabeled    | 58.4         | 52.3  | 54.7  | 54.1   |
| Named Ent.   | -14.4        | -1.1  | -0.3  | 31.8\(^†\) |
| Negation     | -2.00        | -0.0  | 3.4   | 29.0\(^†\) |
| Concepts     | 76.7\(^†\)  | 64.5  | 72.3  | 68.0   |
| Coreference  | 23.2         | 10.3  | 13.6  | 21.2   |
| S\(\)L       | 48.3         | 40.8  | 44.9  | 50.0\(^†\) |
| maxIndegreeSim | 27.0     | 23.6  | 24.0  | 26.4   |
| maxOutDegreeSim | 22.3     | 17.5  | 19.4  | 23.1   |
| maxDegreeSim | 22.3         | 18.0  | 19.7  | 22.5   |
| rootSim      | 25.5         | 21.7  | 25.1  | 28.9\(^†\) |
| quantSim     | 11.5         | 10.0  | 11.8  | 65.4\(^†\) |

Table 1: Spearman \(\times 100\) of AMR aspects. *Italics*: overall best. **bold**: best partitioning approach. underline: improvement by more than 20 Spearman points.

where the weight \(\omega(i, j)\) is given as the Spearman correlation that is achieved when comparing the SBERT values of dimension \(i\) against the metric \(\rho\) against the metric score for aspect \(j\) across all (development) data instances. The partitioning is achieved by computing

\[
\max \sum_{(i,j) \in V_{SB} \times V_{SEM}} \omega(i, j) \cdot x_{ij} \quad \text{(7)}
\]

subject to:

\[
\sum_j x_{ij} \leq 1 \forall i \in V_{SB} \quad \text{(8)}
\]

and

\[
\sum_i x_{ij} \geq 1 \forall j \in V_{SEM} \quad \text{(9)}
\]

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** The results of this experiment are displayed in Table 1. We make the following observations: First, the general AMR metrics that measure the graph similarity globally (S\(\)\(^3\)MATCH, W\(\)LKERNEL, W\(\)LKERNEL) are best modeled with the cosine distance computed on full SBERT embeddings (unpartitioned, Table 1). We can’t model them well with a sub-embedding. This seems intuitive: the power of a low-dimensional sub-embedding is too low to express the complex global metrics well. Nonetheless, compared to the best baseline that performs space partitioning (SB-ILP), our approach provides substantial improvements (S\(\)\(^3\)MATCH +2.3
Spearmanr points, \textsc{W.LKernel} +6.0, \textsc{W.WLKernel} +3.2).

Therefore, second, it is more interesting to study the fine-grained semantic aspects measured by our aspectual AMR metrics. We find that there are three AMR features that are barely modeled with global SBERT embeddings: \textit{named entities}, \textit{negation}, \textit{quantification}. They also cannot be properly represented by the SB-ILP baseline. By contrast, S$^3$BERT clearly improves over these baselines. E.g., the \textit{negation} modeling improves from a negative correlation to a significant positive correlation of 29.0 Spearmanr. \textit{Quantifier similarity} increases from 11.8 Spearmanr to 65.4.

5.2 Correlation with human judgements

Relating to research question 2): 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 legitimate objective.

5.2.1 Sentence similarity

Testing data We use sentence similarity data sets with human ratings, where the STS benchmark (STSb) assesses sentence similarity (Baudiš et al., 2016b) and SICK sentence relatedness (Marelli et al., 2014). We min-max normalize the Likert-scale ratings of both datasets to the range between 0 and 1.

Evaluation metric We again use Spearmanr. To assess efficiency, we display the approximate time needed to execute the metric over 1,000 pairs. We also want to assess the \textit{explainability} of the methods, which can be complicated (Danilevsky et al., 2020). To keep it as simple as possible, we assign \texttt{**} if a metric is fully transparent, and the score can be traced (e.g., via an AMR-to-AMR alignment of SMATCH or \textsc{W.WLKernel}), and one \texttt{*} 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 \texttt{deco}) or b) the consistency objective (S$^3$BERT \texttt{cons}). 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).

Results are shown in Table 2. Interestingly, while one of our concerns was the risk of catastrophic forgetting, S$^3$BERT outperforms all baselines, including the SBERT model. On STSb the performance increase is significant. This shows that the consistency loss was effective in preventing the anticipated catastrophic forgetting. Without the consistency loss, the performance drops by about 10 points (S$^3$BERT \texttt{cons} in Table 2).

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: \textit{dissimilar}, \textit{unrelated}, \textit{what similar} and \textit{highly similar}. Originally, the task was evaluated as a binary classification task (Reimers et al., 2019), by mapping the \textit{similar} and \textit{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 the prior work, we also evaluate using this setup. However, to account for the fine-grained labels on a three point Likert

\begin{table}[h]
\centering
\begin{tabular}{|l|c|c|c|c|}
\hline
system & \multicolumn{2}{|c|}{xplain} & \multicolumn{2}{|c|}{Spearmanr x 100} \\
      & 3-Likert & binary classifier & F1 scores & \\
      & Macro & Sim. & Sim. & \\
\hline
RE19   & -       & -       & 65.4 & 52.3 & 78.5 \\
BH21   & -       & 34.8 & -   & -   & -   \\
OP21   & \texttt{**} & -   & 68.6 & 60.4 & 77.0 \\
SBERT  & -       & 54.2 & 71.7 & 63.8 & 79.6 \\
S$^3$BERT & \texttt{†} & \textbf{56.4} & 72.9 & \textbf{65.7} & \textbf{80.1} \\
\hline
\end{tabular}
\caption{Table 3: Results on argument similarity prediction.}
\end{table}
scale (3-Likert), we also use a second metric based on correlation, following a proposal by Behrendt and Harmeling (2021): We map dissimilar and unrelated to 0, somewhat similar to 0.5, highly similar to 1.0. We then compute Spearman.

**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 metric approach of Opitz et al. (2021b) (OP21), and iii) Spearman correlation as in Behrendt and Harmeling (2021) (BH21), who pre-train BERT on more 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 setting. When assessing interpretability, OP21 offers because it is based on SMATCH and the score can be fully traced (details in the Appendix A.2). However, it is less efficient, due to the cost of executing AMR metrics and parser, and lags behind in accuracy. Hence, we can conclude that our approach offers a valuable balance between interpretability and performance.

6 Analysis

6.1 Interpreting $S^3$BERT predictions
We find interesting cases where $S^3$BERT is able to explain its similarity scores. For example, both $S^3$BERT and SBERT assign a high similarity score (.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 .09 in $S^3$BERT’s quantSim partition for quantifier meaning provides a (possible) rationale for the much lower human score, due to a strong contrast in quantifier meaning (*two* vs. *a*).

An interesting orthogonal case is where we find a rationale for a model rating that explains its score possibly better than the human rating, which might indicate that a different human label could be appropriate. The human similarity rating for *a man is smoking and a baby is sucking on a pacifier* is at minimum (zero), indicating complete dissimilarity. $S^3$BERT, however, assigns a score that indicates a small similarity, and its decomposed embeddings justify its decision: they show that there is no contradiction in negation (0.87), the focus of the sentence is somewhat related (0.6), and the frames (*sucking* vs. *smoking*) are judged to bear some low similarity (0.21).

6 See more examples in Appendix 4.

Figure 2: STSb vs. argument similarity (UKPA) when viewing similarities of aspectual sub-embeddings against human ratings.

6.2 Similarity vs. Argument Similarity
We can view $S^3$BERT’s similarity predictions in selected semantic aspects and assess how they relate to human annotator similarity in rating argument similarity vs. sentence similarity.

According to Figure 2 the distribution of aspects is amazingly similar between the two datasets. The most notable difference can be found in quantifier phenomena, which seem to influence human ratings in STSb more than in UKPA. We speculate that for argument similarity users tend to generalize more over different quantifiers since they tend to be more oriented to general statements as opposed to, e.g., exact numerical differences.

The greater influence of the residual embedding for rating argument similarity may indicate that the argumentation data may be more complex and one has to go beyond what the AMR metrics can measure. This can be explained by the fact that argumentation involves controversial political topics, which increases the relevance of background knowledge, to properly assess similarity between arguments. Such knowledge may – to some extent – be contained in $S^3$BERT, but it is not captured by its meaning features. We will investigate the inclusion of more explicit background knowledge in future work.

7 Conclusion

With the goal to obtain sentence similarity ratings that are both accurate, efficient and explainable, we propose a method that structures the neural sentence embedding space into different sub-spaces. The sub-spaces express linguistic facets of meaning as captured by AMR and AMR metrics, such as Negation or Semantic Roles. Our decomposition objective performs the semantic space partitioning via targeted synthesis of an AMR metric ensemble. During the process, guided by our consistency objective, a residual embedding retains valuable
information semantic information that cannot be expressed by AMR and AMR metrics.

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 Šedivý. 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. Argumentation as explanation when we have saliency methods. 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.

Daniel Cer, Mona Diab, Eneko Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 1–14, Vancouver, Canada. 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 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.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yanis 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.

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 468–483, 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, Premtih Samhataj, Sean Kallenberg, Christopher Coors, Lorik Dumanli, 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.

Viswanath Nagarajan and Maxim Sviridenko. 2009. On the maximum quadratic assignment problem. Mathematics of Operations Research, 34(4):859–868.

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.

Edwina L Rissland, David B Skalak, and M Timur Friedman. 1993. BankxX: a program to generate argument through case-base search. In Proceedings of the 4th international conference on Artificial intelligence and law, pages 117–124.

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, MichaelCogswell, 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 Bili, 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 Linares-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 An-notation 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. *arXiv preprint arXiv:2205.12176*.

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*.

### Example 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 Smatch for argument similarity rating

Opitz et al. (2021b) create $S^2$-match (Opitz et al., 2020) variants to achieve explainable argument similarity. The variant shown in our results achieves state-of-the-art argument similarity rating scores by biasing Smatch score and alignment such that concept node matches are assigned triple weight, discounting graph structure triple matches.

#### A.3 Prediction examples
### Table 4: Example analysis from STSb.

| sentence A                          | sentence B                          | humSim | S^BERT | notable feature similarities |
|-------------------------------------|-------------------------------------|--------|--------|------------------------------|
| two cats looking at a window        | a white cat looking out of a window | 0.52   | 0.73   | quant: 0.09↓                 |
| a cat is looking at a window        | a cat is looking out of a window    | na     | 0.93   | quant: 0.06↑; SRL: 0.72↓     |
| three men posing in a tent          | three men eating in a kitchen       | 0.24   | 0.42   | quant: 0.8↑                  |
| rocky andapollo creed are running down the beach | the men are jogging on the beach | 0.6    | 0.32   | rootSim: 0.59               |
| a man is smoking                    | a baby is sucking on a pacifier     | 0.0    | 0.10   | Concepts: 0.45; Frames: 0.21↑; Negation: 0.87↑; maxDogSim: 0.67↑ |
| a dog prepares to herd three sheep with horns | a dog and sheep run together | 0.44   | 0.65   | Frames: 0.26↑; Negation: 0.75↑; quant: 0.01↓                 |
| the girl has something on her head  | the woman has something with her    | 0.52   | 0.38   | Concepts: 0.13; Negation: 0.80; Coref: 0.45; SRL: 0.43; SMATCH: 0.12; WlKernel: 0.02; WlKernel: 0.55 |