Similar, but why? A Toolkit for Explaining Text Similarity

Juri Opitz[1*] **Andrianos Michail**[1*] **Lucas Möller**[2] **Sebastian Padó**[2] **Simon Clematide**[1]

[1]University of Zurich, Switzerland
[2]IMS at University of Stuttgart, Germany

[1]{jurialexander.opitz,andrianos.michail,simon.clematide}@uzh.ch
[2]{lucas.moeller,pado}@ims.uni-stuttgart.de

Abstract

Explaining text similarity and developing interpretable models are emerging research challenges (Opitz et al., 2025). We release XPLAINSIM, a Python package that unifies three complementary approaches for explaining textual similarity in an easily accessible way: 1. a token attribution method that explains how individual word interactions contribute to the predicted similarity of any embedding model; 2. a method for inferring structured neural embedding spaces that capture explainable aspects of text, and 3. a symbolic approach that explains textual similarity transparently through parsed meaning representations. We demonstrate the value of our package through intuitive examples and three focused empirical research studies. The first study evaluates interpretability methods for constructing cross-lingual token alignments. The second investigates how modern information retrieval methods handle stop words. The third sheds more light on a long-standing question in computational linguistics: the distinction between relatedness and similarity. XPLAINSIM is available at https://github.com/flipz357/XPLAINSIM.

1 Introduction and Background

Understanding semantic similarity is an important research question, both from an academic and practical perspective (Opitz et al., 2025). Partly, this is likely because it is a challenge in itself to express what makes two texts (dis)similar, even for humans (Fodor et al., 2024). Importantly, among the wide range of explainability methods (Sundararajan et al., 2020; Janizek et al., 2021), explanation of similarity represents a special case, since the assessment critically depends on *interactions between two inputs*, which significantly increases the complexity of interpretation and explanation (Figure 1). For example, when texts are represented as embeddings, the multiplicative interaction of the two inputs through cosine distance yields the similarity score.

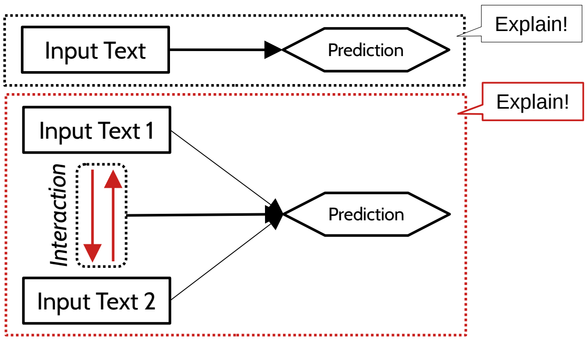

Figure 1: Complexity of explanation. Top: "Classic" single-input-prediction explanation problem. Bottom: A prediction for two inputs is not only influenced by either input, but also, first and foremost, by *their interaction*.

And such questions are not just theoretical. For instance, systems that search texts rely on text similarity—in sensitive domains, like law or medicine, understanding why a system delivers certain texts, but not others, may become especially crucial. Against the background of emerging AI laws (e.g., "right to explanation"; EU, 2024) the demand for transparency is expected to intensify.

In this work, we release an easy-to-use software package that combines three broadly applicable approaches for generating human-interpretable explanations of semantic textual similarity. The target audience covers both developers and end users interested in a better understanding of models of semantic similarity and the interpretation of their output. Finally, our toolbox can be a starting point for research in this area.

In its current version, our package includes three distinct explanation approaches. We select these three approaches since each of them is addressing the problem of similarity interpretability from a complementary perspective aligned with three explanation paradigms (Opitz et al., 2025): **Interaction attribution** (Moeller et al., 2023, 2024) is

[*] Equal contribution.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 203–214
March 24-29, 2026 ©2026 Association for Computational Linguistics

Module	Tooling
`attribution`	Retrieve token-pair interactions for off-the-shelf models
`spaceshaping`	Learn structured neural representation of semantic aspects
`symbolic`	Meaning Representation similarity/ Parsing and matching graphs

Table 1: Simplified overview of included modules.

a *post-hoc approach* that allows to investigate off-the-shelf embedding models by explaining their decisions from an input perspective, focusing on token interactions. **Space shaping** (Opitz and Frank, 2022) lets us create *interpretable embeddings*. It enables the integration of custom semantic aspects into an embedding model, supporting fine-grained analysis of semantic decisions as well as efficient clustering and search. Finally, our **symbolic approach** (Banarescu et al., 2013; Opitz, 2023) uses Abstract Meaning Representation (AMR) graphs and grounds the similarity in the comparison of these *structured objects*.

An overview of all three available approaches is provided in Table 1. The modular design of our package supports future extensions and the integration of additional interpretability methods.

Dependencies & License. XPLAINSIM is released under the GPLv3 license and is publicly available at `https://github.com/flipz357/XPLAINSIM`. It can be installed via `pip` for Python and uses the import namespace `xplain` for brevity. Its main dependencies are `pytorch`[1] and `sentence-transformers`[2]. For symbolic similarity computation, we additionally rely on `amrlib`[3] and `smatchpp`[4].

Research study contribution. In addition, our paper highlights our package's value by contributing three focused research studies. The first experiment investigates the intrinsic cross-lingual alignment capabilities of multilingual embedding models. The second examines how IR-focused models handle stop words when matching queries to candidate documents. Both of these underlying phenomena are not directly observable in text embedding models and only become accessible through explainability methods. The third study uses measurements grounded in symbolic representations to shed light on structural differences between similarity and relatedness, two notions often conflated. These pilot studies demonstrate the practical benefits and distinct capabilities of our toolkit.

```
from xplain.attribution import ModelFactory
model= ModelFactory.build("gte-multilingual-base")
a= 'The dog runs after the kitten in the yard.'
b= 'Im Garten rennt der Hund der Katze hinterher.'
raw_A, tokens_a, tokens_b = \
    model.explain_similarity(a, b)
A_pp, tokens_a_pp, tokens_b_pp = \
    model.postprocess_attributions(...)
```

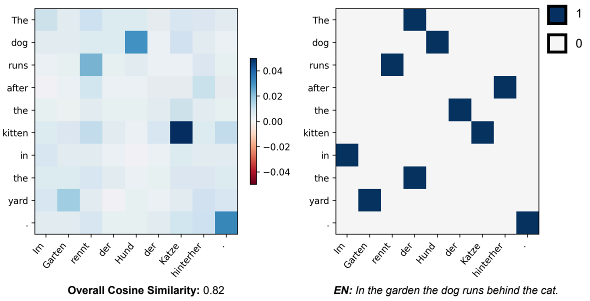

Figure 2: **Top:** Attribution generation for the off-the-shelf text embedding model `gte-multilingual-base`. **Left**: Attribution matrix, whose sum approximates the overall similarity score. **Right**: Sparsified attribution matrix generated by FlowAlign post-processing.

Related work and context. Several interpretability libraries exist for neural network-based NLP models (e.g., Kokhlikyan et al., 2020; Alammar, 2021; Attanasio et al., 2023; Fiotto-Kaufman et al., 2025). However, all these tools are focused on a single-input setting: they explain how input features contribute to a single model prediction. Text similarity is fundamentally different, because any prediction arises not only from each input in isolation but from the *interaction/comparison* between two inputs (cf. Figure 1). It is this gap that our XPLAINSIM package targets.

2 Implemented Methods

In this section, we introduce each of the three explainability methods included in XPLAINSIM and provide an example outlining its idea and usage.

2.1 Token Interaction Attributions

Explanation mechanism. For a given prediction, attribution methods assign importance values to input features, which, in our case, are pairs of words. That is, every pair of words (across two inputs) is

[1] https://github.com/pytorch/pytorch

[2] https://github.com/UKPLab/sentence-transformers

[3] https://github.com/bjascob/amrlib

[4] https://github.com/flipz357/smatchpp

assigned a weight and the sum of all weights approximates the overall similarity score assigned by the embedding model. Moeller et al. (2023; 2024) have developed methods that compute such weights for embedding models. The interaction attributions generated by their method can be visualized as a matrix that decomposes the overall similarity score (cf. the left matrix in Figure 2).

Implementation. The method builds on the concept of integrated gradients, extending it to Siamese encoder architectures such as sentence transformers (Reimers and Gurevych, 2019). The resulting attributions are based on the *integrated Jacobians* of the embeddings of the two inputs with respect to their token representations. The Jacobians are computed with automatic differentiation; technical details can be found in the original publications.

We integrate the existing code base into our package and extend it in two ways: (1) We add support for multilingual and retrieval-focused embedding models. (2) We integrate optional post-processing for discretizing attribution matrices. This is motivated by the observation that raw attribution matrices tend to be relatively flat, making interpretation difficult. Discretization highlights the most relevant token-to-token alignments. These extensions provide the foundation for two demonstration experiments presented in Section 3.

Post-attribution alignment sparsification. The resulting attribution matrices indicate correspondences between the two inputs, similar to alignment matrices commonly used in machine translation. To enable comparison with alignment annotations, these matrices must be discretized and converted into a sparse binary format (Dou and Neubig, 2021). We implement two strategies:

MaxAlign: Each token is aligned to its strongest counterpart in the other sentence, but only if the link is mutual. Let $\text{AtoB}(a)$ denote the token b in the second sentence that receives the highest attribution from token a, and $\text{BtoA}(b)$ analogously for token a. Then:

$$\text{Align}_{ab} := \mathcal{I}[\text{AtoB}(a) = b \land \text{BtoA}(b) = a].$$

where $\mathcal{I}[s]$ returns 1 if s is true, and 0 otherwise. This strategy yields a sparse, precision-oriented alignment by linking only mutually preferred token pairs and ignoring ambiguous or weak links.

FlowAlign: We also provide a more advanced alignment based on optimization. The attribution matrix is interpreted as a cost matrix by transforming attribution scores into costs (higher attribution means lower transport cost). Each token is assigned a weight of 1, and the Wasserstein distance is computed (aka Earth/Word Mover's Distance, minimal transport Kusner et al., 2015). Intuitively, this distance expresses the minimal amount of work required to transform one set of embeddings to the other. A by-product of this computation is a sparse *Flow* matrix between embeddings (transportation plan), which we use as alignment $\text{Align}_{ab} := \mathcal{I}[\text{Flow}_{ab} > \tau]$, where τ is a threshold.[5] The resulting alignment is slightly less sparse (n:m) and yields higher recall compared to MaxAlign.

Example. Figure 2 illustrates the process. We compute the similarity between an English and a German sentence using a multilingual GTE model (Li et al., 2023). Our code then generates the corresponding attribution matrix. Finally, we apply FlowAlign to discretize the attributions, revealing that the model aligns cross-lingual word pairs such as *dog – Hund*, or *kitten – Katze*. A systematic evaluation of this behavior is presented in Section 3.

2.2 Space Shaping

Explanation mechanism. Text embeddings are points in a high-dimensional vector space. This explanation method aims to decompose that space into lower-dimensional subspaces, each capturing a distinct semantic aspect (Opitz and Frank, 2022). For example, one subspace might represent named entities mentioned in a text, while another captures topical content. Similar to the attribution method, this approach decomposes the overall similarity score into multiple contributions—but at the level of abstract semantic aspects rather than individual token interactions. Such a decomposition enables explanations like: "two texts are similar in aspect X but differ in aspect Y". In contrast to local attribution methods, which compute explanations during or after the similarity calculation, this approach embeds the explanation directly into the model. As a result, all explanatory structure is learned during training, and explanations are available at inference time without additional computational cost.

[5] An intuitive choice is $\tau = 0$, though we found that $\tau = 0.029$ performs best across all language pairs in the dev set of the alignment task, and we set this as the default.

```
from sentence_transformers import InputExample
from xplain.spaceshaping import \
  PartitionedSentenceTransformer

# we need two lists with documents pairs
docs1, docs2 = ["abc",....], ["xyz",...]

# compute the training/partitioning signal
examples = []
for x, y in zip(docs1, docs2):
    similarities = []
    for metric in my_custom_metrics:
        similarities.append(metric.score(x, y))
    examples.append(InputExample(texts=[x, y], \
            label=similarities))

# instantiate model, we use 16 dimensions
# to express each metric
pt = PartitionedSentenceTransformer(
   feature_names=[metric.name for \
      metric in my_custom_metrics],
   feature_dims=[16]*len(my_custom_metrics))
train_examples, dev_examples = split(examples)
# train partitioning
pt.train_model(train_examples, dev_examples)
```

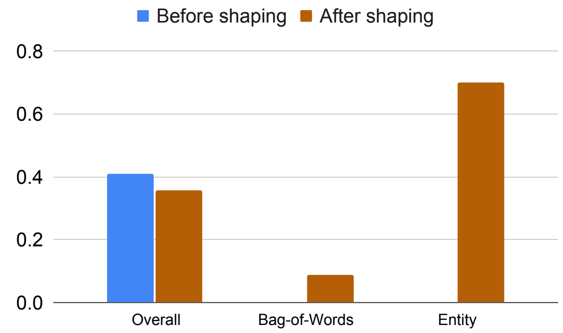

Figure 3: **Top:** Pseudo code to induce semantic subspaces. **Bottom:** Aspect similarities for the sentences "The kitten drinks milk" vs. "A cat slurps something."

Implementation. Two challenges arise: Learning to partition the space, and preventing that the overall similarity of two texts deviates too much from a strong reference model[6].

We extend the "S3BERT" method[7] from Opitz and Frank (2022) which used AMR-based metrics to measure aspectual similarities and generate a partitioning signal. We generalize this approach to enable the shaping of fully custom embedding spaces. Users can define their own interpretable similarity metrics to create distinct embedding subspaces. In the following paragraph, we provide an example of the training and inference process.

```
from xplain.symbolic.model import AMRSimilarity
explainer = AMRSimilarity()
sent1 = ["Barack Obama holds a talk"]
sent2 = ["Hillary Clinton holds a talk"]
exp = explainer.explain_similarity(sent1, sent2)
# yields aspectual similarities:
# 'AGENT':    71.43, # (Ob. vs Cl. talking)
# 'NER':      60.0,  # (entity: !=, type: ==)
# ...,        ...,   # (various other stats)
# 'POLARITY': 100.0  # (both: no negation)
# 'global':   80.0   # (overall similarity)

# ::snt Barack Obama holds a talk.
(h / hold-04
      :ARG0 (p / person
            :name (n / name
                  :op1 "Barack" :op2 "Obama" ))
      :ARG1 (t / talk-01))

# ::snt Hillary Clinton holds a talk.
(h / hold-04
      :ARG0 (p / person
            :name (n / name
                  :op1 "Hillary" :op2 "Clinton") )
      :ARG1 (t / talk-01))
```

Figure 4: **Top:** Using XPLAINSIM's symbolic part to explain similarity via meaning graph difference statistics. **Bottom:** AMR graph differences make meaning disagreement overt.

Example. Figure 3 gives a high-level view of the space shaping process (the base model here is all-MiniLM-L12-v2, Reimers and Gurevych, 2019). First, the user defines interpretable aspects of interest using custom metrics. These metrics can be simple and approximate—for example, measuring word overlap between two texts to capture superficial structural similarity, or measure the similarity in named entity structure of the two input texts via SpaCy[8] NER tagging. A set of paired training texts is then created by assigning scores based on these custom metrics.

2.3 Symbolic Explanation with AMR

Explanation mechanism. A meaning representation (MR) is a symbolic encoding of the semantic structure of a text, typically grounded in linguistic theories of compositionality and discourse. An MR can take the form of a graph, where nodes represent entities and events mentioned in a text, and edges show their semantic relations (*agent*, *patient*, *instrument*, *cause*, etc.). With meaning expressed in such an explicit format, a metric between MRs can highlight (dis-)agreements with respect to specific parts and properties of semantic structure. Several

[6] Such a reference model can be any embedding model, from which the explainable partitioned model learns to maintain high prediction accuracy.

[7] https://github.com/flipz357/S3BERT

[8] https://spacy.io/

papers have already explored MR measurements for, e.g., semantic similarity, language inference, generation evaluation, or cross-lingual analysis.[9] We provide the first ready-to-use tool that integrates parsing and aspect-based AMR similarity measurement in a single package. Ready-to-use specifically means that a strong default parser is integrated in our package, which is, to our knowledge, not the case for any other MR metric package.

Implementation. We need to select a type of meaning representation, a parser that generates such representations, and a metric that can compare such representations as well as (ideally) any of their subgraphs. For our package, we also offer a default way of measuring, based on two pillars:

1. Representation and Parsing: The AMR representation (Abstract MR, Banarescu et al., 2013) has broad applications and large resources (Wein and Opitz, 2024; Sadeddine et al., 2024) as well as fairly accurate parsers (Bai et al., 2022). Concretely, we leverage the `amrlib` library that has pre-trained parsers for AMR representation. For an overview of the currently available parsing models we refer the reader to the Appendix, Table 5.

2. Measuring: Finding the largest common subgraph of two AMR graphs is an NP-complete problem (Allen et al., 2008; Nagarajan and Sviridenko, 2009; Cai and Knight, 2013). We adopt the `smatchpp` library (Opitz, 2023), a graph matching library that uses Integer Linear Programming. The similarity score of two graphs is the amount of shared nodes plus the amount of shared edges, normalized by the size of either graph, obtaining two directional similarities. For a symmetric similarity score, the directional similarities are averaged with harmonic mean. We compute this measure for several aspects elicited by AMR subgraphs, e.g., coreference, negations, named entities. For an overview of the currently available measurements we refer the reader to the Appendix, Table 4.

Example. Figure 4 shows an application of the AMR-based approach to two sentences. After parsing the inputs and measuring aspectual similarities, we find that both sentences have a similar event with a different agent (`arg0`). Matching the full graphs with `smatchpp` yields a similarity of 0.8. Matching only the subgraphs capturing patients in events (`arg1`), yields a perfect match (essentially: Someone holds a talk), but the subgraphs that capture Named Entities match with only 0.6 (indeed, their only agreement is the type (`person`)).

Model	Discretiz.	X-Ling Word Alignment		
		Pr	Re	F1
Baseline	Diagonal	0.262	0.245	0.253
XLM-R	MaxAlign	0.527	0.064	0.114
	FlowAlign	0.520	0.470	0.494
M-MPNet	MaxAlign	0.797	0.334	0.471
	FlowAlign	0.636	0.578	0.606
M-MiniLM	MaxAlign	**0.799**	0.371	0.507
	FlowAlign	0.662	0.605	0.632
M-E5-Base	MaxAlign	0.775	0.477	0.591
	FlowAlign	0.668	**0.610**	**0.638**
M-GTE	MaxAlign	0.779	0.444	0.566
	FlowAlign	0.667	**0.610**	0.637

Table 2: Cross-lingual word-level alignment results aggregated across all languages. Full model ID's: Table 3

3 Pilot Studies

We show the potential utility of our XPLAINSIM package in three empirical pilot studies.

3.1 Cross-lingual Alignment

Multilingual embedding models are typically trained contrastively, on positive and negative pairs across languages, but are not explicitly supervised at the word level. In this experiment, we investigate to what extent multilingual models internally develop unsupervised cross-lingual word alignments.

Experiment. We use the ten English–{bg, da, es, et, hu, it, nl, pt, ru, sl} gold alignment datasets released by Martelli et al. (2023), see Appendix C for details. For each sentence pair, we compute interaction attributions from several multilingual embedding models. To compare these attributions with the gold alignment annotations, we discretize them into binary matrices. For this purpose, we apply both of our sparsification methods: the row- and column-wise max-pooling heuristic (MaxAlign), and the optimal transport-based approach. We evaluate the resulting binary alignment matrices using standard precision, recall, and F1 score. As a baseline, we include a diagonal alignment matrix.

[9] For a sample, we refer the reader to this list: Manning and Schneider (2021); Opitz and Frank (2021); Wein and Schneider (2024); Müller and Kuwertz (2022); Opitz et al. (2023); Ghosh et al. (2024); Jayaweera et al. (2024); Kachwala et al. (2024); Sun and Xue (2024); Landes and Di Eugenio (2024); Park et al. (2024); de Vergnette et al. (2025); Thatikonda et al. (2025)

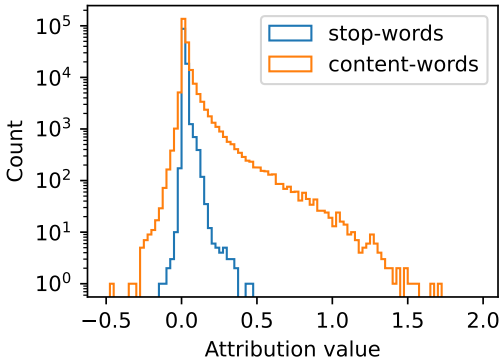

Figure 5: Histograms of attributions to stop and content words; MS MARCO validation split, IR model.

Result. Table 2 shows that M-GTE and M-E5-Base perform best in approximating a human-style discrete alignment. Notably, these models also perform well on multilingual retrieval data (e.g., MIRACL Zhang et al., 2023), suggesting that stronger alignment capabilities contribute to high performance in multilingual similarity and retrieval tasks.

FlowAlign post-processing consistently improves alignment performance across models. For users aiming to extract word-level alignments from attributions, we recommend combining a strong embedding model with FlowAlign for sparsification. In some cases, this combination yields substantial gains, e.g. XLM-R improves from an F1 score of 0.11 to 0.49, a 38-point increase. The impact of sparsification becomes smaller for higher-performing models, suggesting that stronger models rely on a latent discrete alignment.

3.2 Stop-words in Dense Retrieval

Semantic search is an application of text embedding models with high practical relevance. Dual encoder architectures can process queries and documents independently. The storage of document embeddings in vector databases enables efficient search in sublinear time complexity via approximate nearest neighbor algorithms. However, compressing entire documents into unified vector representations makes such dense retrieval results difficult to interpret. Using the interaction attribution method by Moeller et al. (2023), we study which parts of a given query and document a model matches.

Experiment. We build on the common distinction between stop and content words to evaluate whether an IR-optimized text embedding model effectively learns to suppress stop words. To this end, we use the MS MARCO Passage Ranking dataset (Bajaj et al., 2016) (details in Appendix C) and evaluate a model fine-tuned on its training split[10]. For each positive query–passage pair in the validation split, we compute interaction attributions and sum the total attribution to all passage tokens. We then aggregate these contributions separately for stop words and content words, using NLTK's stop word list for categorization.

Result. Figure 5 shows histograms of all stop- and content-word attributions, respectively. 97.6% of all attributions to stop-words fall within an interval of ± 0.05 around zero. This shows that the model learned to effectively ignore stop words. The distribution for content words, on the other hand, is much wider, ranging from -0.46 to 1.70. The model assigns both strongly positive and negative importance scores to words, suggesting that it can reward well-aligned document segments, penalize mismatches, and ignore large portions of document content, as indicated by the many content words whose attribution is near zero. Understanding which parts of queries and documents such models align—and where they are prone to errors or biases—is a promising direction for future research.

3.3 Relation Characterization with a Symbolic Approach

"Similarity" and "relatedness" are often treated as equivalent notions; however, this equivalence is not entirely accurate, as latent differences exist between them (Budanitsky and Hirst, 2006). To better understand the structural distinctions between similarity and relatedness, we employ symbolic, AMR-based aspectual measurements.

Experiment. For the similarity dataset, we use the validation partition of the STS benchmark (Cer et al., 2017); for relatedness, we use the SICK dataset (Marelli et al., 2014). Both datasets are well-established evaluation benchmarks, see Appendix C for details. We compute aspectual graph metrics for all sentence pairs in both datasets. Correlating these measures directly with the human similarity score is confounded by cases where particular semantic aspects are absent in both texts. Therefore, we compute Pearson correlation separately for each aspect, considering only sentence pairs where the graph metric detected an aspectual

[10] sentence-transformers/msmarco-MiniLM-L12-cos-v5

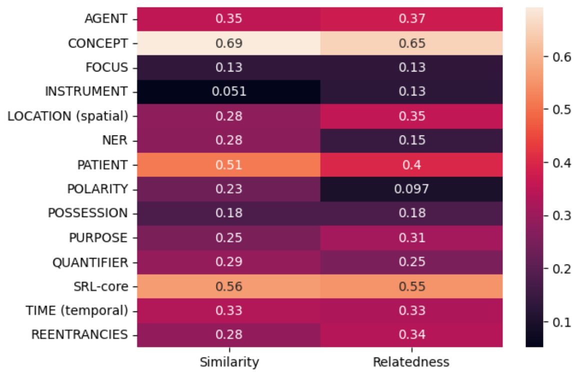

Figure 6: Pearson correlation of aspectual differences, captured via symbolic meaning (sub-)graphs and graph metrics, with human-assessed similarity (STS) and relatedness (SICK).

difference.

Result. Figure 6 presents the obtained Pearson correlations. A notable difference is that variations in polarity and named entities are predictive of differences in human-assessed similarity (STS), but less so for relatedness (SICK). A commonality across both similarity and relatedness is their sensitivity to differences in shared concepts. Interestingly, similarity differences are more strongly associated with changes in *patients* (entities undergoing an action), whereas relatedness appears more influenced by the similarity of *agentive* structures.

Discussion. This analysis provides a rich contrastive characterization of similarity and relatedness. It also extends prior work, which has largely focused on individual words, by examining full semantic structures instead.

4 Summary

We release the XPLAINSIM package to support a better understanding of semantic similarity, and how it is computed in neural and non-neural models. In particular, we provide three types of methods, each with distinct trade-offs. Attribution-based methods reveal how interactions between words contribute to the final similarity score, from the perspective of neural embedding models. The most efficient approach is space shaping, which integrates explanations directly into the embedding space, effectively reducing the cost of generating explanations to zero. While it requires model training, it also enables customization of the explanation. Finally, the symbolic approach offers fine-grained comparisons based on meaning representations, adding an additional layer of transparency. Its limitations include reduced sensitivity to lexical nuance and dependence on parsing components.

Within this package description paper, we also contribute three research studies that highlight the value of the tools: 1. evaluating cross-lingual token alignment, 2. assessing structural token-weighting in information retrieval models, and 3. examining the distinction between similarity and relatedness.

We hope that XPLAINSIM lowers the barrier for both researchers and practitioners to explore explanations for why two texts are similar, beyond a single similarity score.

Limitations

Our package is designed to be broadly applicable for all kinds of text similarity and explanation tasks. Potential constraints may arise from the specific characteristics of the included models: Space shaping requires custom design of measures and training, attribution analysis is applicable to the broader class of Siamese transformer models but comes at high computational costs that make it infeasible to apply to long-context tasks without further adaptations. Finally, meaning representations and their metrics provide highly interpretable and controllable similarity statistics, but they currently may not correlate as highly with human similarity ratings in STS annotation studies when compared to large neural models. This may not only be due to some structural insufficiencies, but also potentially due to noise from the parsing process, further aggravated by eventual domain shifts.

However, we emphasize that our package is designed to be easily usable, and readily extensible, such that implementation of other potential approaches, variation or improvement of current ones, and adding functionality, is straightforward. For future work, we also plan on expanding the library to include other interpretability methods adapted to semantic similarity, e.g., based on causal probing, counterfactual explanations, or Shapley values. We also plan on running human-centered studies for evaluating similarity with and through similarity interpretability methods.

For these and other future works with XPLAINSIM, we warmly invite the community to participate in contributing.

Ethics statement

We do not identify direct ethical risks arising from the release of this toolkit. On the contrary, the primary purpose of our package is to explain models and model decisions. Since models from this domain are widely applied, also in document search contexts, we believe that better understanding their mechanism can only help to also better assess their risks, such as potential retrieval biases favoring certain documents.

Acknowledgments

The Zurich authors received funding from the Swiss National Science Foundation (SNSF 213585) and the Luxembourg National Research Fund (17498891) through the *Impresso II* project.

References

J Alammar. 2021. Ecco: An open source library for the explainability of transformer language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 249–257, Online. Association for Computational Linguistics.

James F. Allen, Mary Swift, and Will de Beaumont. 2008. Deep semantic analysis of text. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 343–354. College Publications.

Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. 2023. ferret: a framework for benchmarking explainers on transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 256–266, Dubrovnik, Croatia. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

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.

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

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 Agirre, Iñigo Lopez-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.

Rémi de Vergnette, Maxime Amblard, and Bruno Guillaume. 2025. Evaluation framework for layered meaning representation. In *Proceedings of the Sixth International Workshop on Designing Meaning Representations*, pages 38–48, Prague, Czechia. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

EU. 2024. Article 86: Right to explanation of individual decision-making.

Jaden Fiotto-Kaufman, Alexander R. Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, Dmitrii Troitskii, Michael Ripa, Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla Brodley, Arjun Guha, Jonathan Bell, Byron C. Wallace, and David Bau. 2025. Nnsight and ndif: Democratizing access to open-weight foundation model internals. In *The Thirteenth International Conference on Learning Representations*.

James Fodor, Simon De Deyne, and Shinsuke Suzuki. 2024. Compositionality and sentence meaning: Comparing semantic parsing and transformers on a challenging sentence similarity dataset. *Computational Linguistics*, pages 1–52.

Sreyan Ghosh, Utkarsh Tyagi, Sonal Kumar, Chandra Kiran Evuru, Ramaneswaran S, S Sakshi, and Dinesh Manocha. 2024. ABEX: Data augmentation for low-resource NLU via expanding abstract descriptions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 726–748, Bangkok, Thailand. Association for Computational Linguistics.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22(104):1–54.

Chathuri Jayaweera, Sangpil Youm, and Bonnie J Dorr. 2024. AMREx: AMR for explainable fact verification. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 234–244, Miami, Florida, USA. Association for Computational Linguistics.

Zoher Kachwala, Jisun An, Haewoon Kwak, and Filippo Menczer. 2024. REMATCH: Robust and efficient matching of local knowledge graphs to improve structural and semantic similarity. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1018–1028, Mexico City, Mexico. Association for Computational Linguistics.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *Preprint*, arXiv:2009.07896.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Paul Landes and Barbara Di Eugenio. 2024. CALAMR: Component ALignment for Abstract Meaning Representation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2622–2637, Torino, Italia. ELRA and ICCL.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

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'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Federico Martelli, Andrei Stefan Bejgu, Cesare Campagnano, Jaka Cibej, Rute Costa, Apolonija Gantar, Jelena Kallas, Svetla Peneva Koeva, Kristina Koppel, Simon Krek, Margit Langemets, Veronika Lipp, Sanni Nimb, Sussi Olsen, Bolette Sandford Pedersen, Valeria Quochi, Ana Salgado, László Simon, Carole Tiberius, Rafael-J. Ureña-Ruiz, and Roberto Navigli. 2023. XL-WA: a gold evaluation benchmark for word alignment in 14 language pairs. In *CLiC-it*.

Lucas Moeller, Dmitry Nikolaev, and Sebastian Padó. 2023. An attribution method for Siamese encoders. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15818–15827, Singapore. Association for Computational Linguistics.

Lucas Moeller, Dmitry Nikolaev, and Sebastian Padó. 2024. Approximate attributions for off-the-shelf Siamese transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2059–2071, St. Julian's, Malta. Association for Computational Linguistics.

Almuth Müller and Achim Kuwertz. 2022. Evaluation of a semantic search approach based on amr for information retrieval in image exploitation. In *2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE.

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

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

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 and Anette Frank. 2022. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *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, Online only. Association for Computational Linguistics.

Juri Opitz, Lucas Moeller, Andrianos Michail, Sebastian Padó, and Simon Clematide. 2025. Interpretable text embeddings and text similarity explanation: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22314–22330, Suzhou, China. Association for Computational Linguistics.

Juri Opitz, Shira Wein, Julius Steen, Anette Frank, and Nathan Schneider. 2023. AMR4NLI: Interpretable and robust NLI measures from semantic graphs. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 275–283, Nancy, France. Association for Computational Linguistics.

Jinwoo Park, Hosoo Shin, Dahee Jeong, and Junyeong Kim. 2024. Improving the representation of sentences with reinforcement learning and amr graph. In *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4.

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.

Zacchary Sadeddine, Juri Opitz, and Fabian Suchanek. 2024. A survey of meaning representations – from theory to practical utility. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2877–2892, Mexico City, Mexico. Association for Computational Linguistics.

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1052–1062, Torino, Italia. ELRA and ICCL.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. 2020. The shapley taylor interaction index. In *International conference on machine learning*, pages 9259–9268. PMLR.

Ramya Keerthy Thatikonda, Wray Buntine, and Ehsan Shareghi. 2025. Assessing the sensitivity and alignment of FOL closeness metrics. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16775–16785, Suzhou, China. Association for Computational Linguistics.

Shira Wein and Juri Opitz. 2024. A survey of AMR applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6856–6875, Miami, Florida, USA. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2024. Assessing the cross-linguistic utility of abstract meaning representation. *Computational Linguistics*, 50(2):419–473.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Short Name	Text Embedding Model HF ID	Base Model HF ID
XLM-R	FacebookAI/xlm-roberta-base	FacebookAI/xlm-roberta-base
M-MPNet	sentence-transformers/paraphrase-multilingual-mpnet-base-v2	FacebookAI/xlm-roberta-base
M-MiniLM	sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	microsoft/Multilingual-MiniLM-L12-H384
M-E5-Base	intfloat/multilingual-e5-base	FacebookAI/xlm-roberta-base
M-GTE	Alibaba-NLP/gte-multilingual-base	Alibaba-NLP/gte-multilingual-mlm-base

Table 3: Model Nomenclature: Mapping of short names to Hugging Face model IDs. The package supports most current embedding models by default. Users can load them by passing the corresponding Hugging Face model ID. Specialized architectures may require the addition of a custom subclass.

Aspect	Brief Description	Trigger Relation(s)	(Typed) Concept Nodes
AGENT	Actor / doer	:arg0	person, animal, nationality ...
CAUSE	Cause of event	:cause	cause-01
CONCEPT	Generic concept	:instance	—
FOCUS	Main predicate	:root	—
INSTRUMENT	Tool used	:instrument	instrument, tool
LOCATION	Place / path	:location, :path, :destination, :direction	city, state, river ...
MATHS	Mathematical entity	—	sum-of, product-of
NER	Named entity	:name	—
PATIENT	Entity affected	:arg1–:arg9	person, object
POLARITY	Negation / polarity	:polarity	—
POSSESSION	Ownership / possession	:poss	owner, possession
PURPOSE	Intended goal	:purpose	purpose-01
QUANTIFIER	Quantity / amount	:quant	monetary-quantity, distance-quantity, volume-quantity ...
QUESTION	Question structures	—	amr-unknown
SRL-core	Core semantic roles	:arg0–:arg9	person, object
TIME (temporal)	Temporal info	:time, :duration, :frequency	date-entity, date-interval
TOPIC	Subject / topic	:topic	topic-01
WIKI	Wikipedia link	:wiki	—

Table 4: Overview of graph aspects, trigger relations, and example concept nodes

A Model Nomenclature

See Table 3.

B Symbolic metrics and parsers

See Table 4 (metrics) and 5 (parsers).

C Dataset Details

Further details on datasets from Section 3: Table 6.

Name	**Size**	**Score**	**Speed**
parse_xfm_bart_large	1.4GB	83.7 SMATCH	17/sec
parse_xfm_bart_base	492MB	82.3 SMATCH	31/sec
parse_spring	1.5GB	83.5 SMATCH	14/sec
parse_t5	785MB	81.9 SMATCH	11/sec
parse_gsii	787MB	76.8 SMATCH	28/sec

Table 5: Default `amrlib` parsing models for which our package automates installation. Table is taken from `amrlib`: `https://github.com/bjascob/amrlib-models`. "Speed is the inference speed on the AMR-3 test set (1898 graphs) using an RTX3090 with num_beams=1 and batch_size=32. The units are sentences/second".

Study	**Dataset**	**Description**	**Size**
Cross-lingual Alignment	XL-WA	Manual word alignment benchmark for 14 language pairs English–X	≈100 sent. in dev set, 200 sent. in test set (1500 word alignments in dev, 4000 word alignments in test) for each language pair. Taken from Martelli et al. (2023)
Stopwords in Retrieval	MS-MARCO (v1.1)	Dataset for IR experiments including real-world questions, gold answers, and a set of candidate passages	9.65K queries in the test split, each with 10 candidate passages. Taken from Bajaj et al. (2016).
Relation Characterization	STS	Sentence pairs with similarity judgments on a Likert scale	1,371 sentence pairs coming from STS Benchmark. Taken from Cer et al. (2017).
Relation Characterization	SICK	Sentence pairs with relatedness judgments on a Likert scale	9,927 sentence pairs coming from SICK-R. Taken from Marelli et al. (2014).

Table 6: Dataset details for the use-case studies.