Title

SemGraphScore: A Novel Metric for Machine Translation Evaluation via Cross-lingual Semantic Graph Matching

Problem Statement

Existing machine translation (MT) evaluation metrics often struggle to capture deep semantic relationships and structural similarities between source and translated texts, especially for languages with vastly different syntactic structures. This limitation can lead to inaccurate assessments of translation quality, particularly for complex sentences or languages with significant structural differences.

Motivation

Current approaches like BERTScore use contextual embeddings for token-level matching, while others like COMET leverage cross-lingual representations. However, these methods don't explicitly model the semantic structure of sentences. By representing sentences as semantic graphs and performing cross-lingual graph matching, we can evaluate translations based on their underlying meaning and relationships, rather than surface-level similarities. This approach has the potential to provide more accurate and interpretable evaluations of translation quality, especially for structurally diverse language pairs.

Proposed Method

We propose SemGraphScore, a novel MT evaluation metric based on cross-lingual semantic graph matching. The method involves five main components: 1) A multilingual semantic parser to convert sentences into abstract meaning representation (AMR) graphs. 2) A cross-lingual graph embedding model that maps semantic graphs from different languages into a shared representation space. 3) A graph matching algorithm that computes similarity scores between source and translated semantic graphs, considering node and edge alignments. 4) A reinforcement learning component that fine-tunes the graph matching process based on human feedback. 5) An interpretability module that highlights specific semantic mismatches or preserved structures in the translation.

Step-by-Step Experiment Plan

Step 1: Data Preparation

Collect parallel corpora for diverse language pairs, including those with significant structural differences (e.g., English-Japanese, English-Arabic, English-German). Use existing MT evaluation datasets like WMT metrics shared task data. Ensure a mix of human references and machine translations.

Step 2: Multilingual Semantic Parsing

Implement or adapt an existing multilingual AMR parser (e.g., AMREAGER or Transition-based AMR parser) to convert sentences into AMR graphs for all languages in the study. Fine-tune the parser on a small set of manually annotated AMRs for each language if necessary.

Step 3: Cross-lingual Graph Embedding

Develop a graph neural network (GNN) model to embed AMR graphs into a shared vector space. Use techniques like graph attention networks or graph convolutional networks. Train the model on parallel AMR graphs to learn cross-lingual representations.

Step 4: Graph Matching Algorithm

Implement a graph matching algorithm that computes similarity scores between source and translated AMR graphs. Consider both node-level and structure-level similarities. Experiment with different matching algorithms (e.g., Hungarian algorithm, graph edit distance) and similarity measures.

Step 5: Reinforcement Learning Component

Design a reinforcement learning framework to fine-tune the graph matching process. Use human judgments of translation quality as rewards. Implement policy gradient methods (e.g., REINFORCE) to optimize the matching algorithm's parameters.

Step 6: Interpretability Module

Develop a module that highlights semantic mismatches and preserved structures in the translation. Use techniques like attention visualization or subgraph isomorphism to identify and display important semantic differences.

Step 7: Baseline Implementation

Implement baseline metrics including BLEU, BERTScore, and COMET for comparison.

Step 8: Evaluation

Evaluate SemGraphScore against baselines on standard MT evaluation datasets. Use correlation with human judgments (e.g., Pearson, Spearman, Kendall's Tau) as the primary evaluation metric. Conduct targeted evaluations on specific semantic phenomena (e.g., negation scope, coreference, idiomatic expressions) to assess the metric's sensitivity to deep semantic transfer.

Step 9: Analysis

Perform detailed analysis of SemGraphScore's performance across different language pairs and semantic phenomena. Use the interpretability module to gain insights into the metric's decision-making process.

Step 10: Ablation Studies

Conduct ablation studies to assess the contribution of each component (semantic parsing, graph embedding, graph matching, RL fine-tuning) to the overall performance.

Test Case Examples

Baseline Prompt Input (BERTScore)

Source (English): The cat is sleeping on the mat. Translation (German): Die Katze schläft auf der Matte.

Baseline Prompt Expected Output (BERTScore)

BERTScore: 0.9532

Baseline Prompt Input (COMET)

Source (English): The cat is sleeping on the mat. Translation (German): Die Katze schläft auf der Matte.

Baseline Prompt Expected Output (COMET)

COMET score: 0.8976

Proposed Prompt Input (SemGraphScore)

Source (English): The cat is sleeping on the mat. Translation (German): Die Katze schläft auf der Matte.

Proposed Prompt Expected Output (SemGraphScore)

SemGraphScore: 0.9721 Interpretation: High semantic similarity preserved. AMR graphs show aligned predicates (sleep-01) and arguments (cat, mat/Matte). Spatial relationship (on) correctly maintained.

explanation

SemGraphScore provides a higher score, indicating better preservation of semantic structure. It also offers an interpretation of the score, highlighting the aligned predicates and arguments in the AMR graphs, which is not available in baseline metrics.

Fallback Plan

If SemGraphScore doesn't outperform existing metrics consistently, we can pivot the project towards an in-depth analysis of where and why semantic graph-based evaluation succeeds or fails. We could focus on specific linguistic phenomena or language pairs where SemGraphScore shows promise, and develop a specialized metric for those cases. Additionally, we could explore combining SemGraphScore with existing metrics in an ensemble approach, leveraging the strengths of both surface-level and semantic-level evaluation. Another direction could be to use the semantic graphs for generating more informative feedback on translation errors, turning the project into a tool for improving MT systems rather than just evaluating them.

Ranking Score: 5