УДК 004.451.25

АНАЛИЗ И ВЫБОР МЕТРИК ДЛЯ ОЦЕНКИ МОДЕЛЕЙ СЕМАНТИЧЕСКОГО СРАВНЕНИЯ: ОТ БАЗОВЫХ ПОКАЗАТЕЛЕЙ ДО РАНГОВЫХ МЕТРИК

Крез К.С., Хлистунова К.В., Кузьмичкина Д.А.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Шнейдеров Е.Н. – к.т.н., доцент, доцент кафедры ПИКС, проректор по учебной работе

Аннотация. В статье систематизированы подходы к оценке моделей семантического сравнения текстов (*Semantic Textual Similarity*, *STS*) и обоснован выбор метрик качества в зависимости от постановки задачи и архитектуры модели. Рассмотрены метрики сходства векторных представлений, показатели бинарной классификации, корреляционные критерии для шкального *STS*, а также ранговые метрики информационного поиска. Показано, что использование единственного показателя (например, корреляции или *Accuracy*) может приводить к некорректным выводам при переносе модели в прикладные сценарии, особенно в двухэтапных конвейерах *retrieval* + *re-ranking*. Предложен практический протокол эксперимента, включающий подбор порога на валидации, расчёт *F*1/*PR-AUC* для верификации и *nDCG@k*/*MRR*/*Recall@k* для задач ранжирования. Приведён пример вычисления метрик на типовых наборах данных и рекомендации по интерпретации результатов с учётом дисбаланса классов, калибровки шкалы и требований к полноте поиска.

Ключевые слова: семантическое сходство текстов; *STS*; метрики качества; эмбеддинги предложений; *cosine similarity*; *Pearson*; *Spearman*; *F1-score*; *PR-AUC*; ранжирование; *nDCG@k*; *MRR*; *Recall@k*; *bi-encoder*; *cross-encoder*; семантический поиск.

Введение. Семантическое сравнение текстов (*Semantic Textual Similarity*, *STS*) – класс задач обработки естественного языка, в которых требуется оценить смысловую близость двух фрагментов текста: предложений, абзацев или документов. В прикладных системах *STS* используется в семантическом поиске, дедупликации, выявлении парафразов, рекомендациях, обработке обращений пользователей, а также в *retrieval*-компонентах вопросно-ответных решений и *RAG*-архитектур, где качество поиска кандидатов напрямую влияет на качество итоговой генерации ответа [5].

Современные подходы к семантическому сравнению в большинстве случаев опираются на векторные представления (эмбеддинги): каждому тексту сопоставляется вектор в многомерном пространстве, а сходство оценивается функцией близости (чаще всего – косинусным сходством). Практическую основу такого направления во многом сформировали *sentence-embedding* модели, построенные по схеме *Siamese*/*bi-encoder*, например *Sentence-BERT* и его последователи [1]. Позднее получили развитие методы контрастивного обучения эмбеддингов, повышающие качество близости векторных представлений на широком спектре задач, включая *STS* и *retrieval* [3], а также текст-к-текст энкодеры, масштабируемые для представления предложений [4].

Однако ключевая проблема прикладной оценки заключается в том, что «качество» модели *STS* нельзя описать одной универсальной метрикой. В зависимости от постановки задачи система может:

– выдавать непрерывную оценку сходства (регрессия по шкале 0-5);

– принимать бинарное решение (парафраз/не парафраз);

– ранжировать множество кандидатов (поиск релевантных документов по запросу);

– сравнивать длинные документы, где сходство носит локальный характер (совпадение отдельных сегментов).

Каждая из этих постановок требует разных метрик и разных протоколов измерения. Например, высокая корреляция предсказаний с человеческими оценками на *STS*-датасете не

гарантирует, что релевантные документы будут попадать в верх выдачи поисковой системы. И наоборот: улучшение *nDCG@*10 может происходить при ухудшении полноты кандидатов (*Recall@k*), что критично для двухэтапных *retrieval + re-ranking* решений.

Цель данной статьи – систематизировать метрики для оценки моделей семантического сравнения, показать ограничения распространённых подходов и предложить практические рекомендации по выбору метрик в зависимости от сценария. В практической части демонстрируется типовой протокол вычисления метрик и интерпретации результатов для задач регрессии, классификации и ранжирования.

Основная часть. Семантическая близость отличается от лексического совпадения. Два текста могут выражать один смысл разными словами (парафраз), а также могут быть лексически похожими, но иметь различный смысл из-за отрицания, перестановки ролей, модальности или контекста. Поэтому «правильная» метрика должна соответствовать тому, как система будет использоваться на практике: сравнение двух конкретных фраз, поиск похожих документов в большом корпусе или оценка степени совпадений в длинном тексте.

Дополнительная сложность состоит в том, что даже «эталон» часто неоднозначен: человеческие оценки сходства субъективны, и разные аннотаторы могут по-разному понимать частичное совпадение смысла. Следовательно, модель в реальности аппроксимирует распределение оценок, а не детерминированную истину. Это особенно заметно в регрессионных задачах *STS*, где требуется согласованность с человеческой шкалой.

В практических системах широко применяются два семейства архитектур:

– *bi-encoder* (двухбашенная схема): каждый текст кодируется независимо, затем вычисляется сходство эмбеддингов (*cosine*/*dot-product*). Преимущество – высокая скорость и возможность индексировать документы заранее, что важно для поиска по большим коллекциям [1,3]. Недостаток – модель не «видит» пару целиком во время сравнения, из-за чего может хуже различать тонкие семантические нюансы.

– *cross-encoder*: пара текстов подаётся совместно, а модель выдаёт релевантность/сходство. Такая схема обычно точнее, но существенно дороже по вычислениям, поэтому её применяют для переранжирования небольшого списка кандидатов (*re-ranking*) после первого этапа *retrieval* [6].

Во многих системах используется двухэтапная стратегия: сначала быстрый *bi-encoder* достаёт *top-k* кандидатов, затем *cross-encoder* уточняет порядок. В этом случае оценка должна отдельно контролировать (а) полноту кандидатов на первом этапе (*Recall@k*), и (б) качество порядка выдачи после ранжирования (*nDCG@k*, *MRR*).

Метрики удобно группировать по тому, какую задачу они оценивают (таблица 1):

– метрики близости векторных представлений (*cosine*, *dot-product*, *L*2);

– метрики бинарной классификации (*Accuracy*, *Precision*, *Recall*, *F*1, *ROC-AUC*, *PR-AUC*);

– корреляционные метрики для шкального *STS* (*Pearson*, *Spearman*, *Kendall*);

– ранговые метрики для *retrieval* (*Recall@k*, *Precision@k*, *MRR*, *MAP*, *nDCG@k*).

Метрики для длинных текстов и сегментов (агрегации по сегментам, покрытия, методы «расстояния между распределениями»).

Таблица 1 – Метрики и корректная область применения

Группа метрик	**Примеры**	**Где применять**
Близость эмбеддингов	*cosine, dot, L2*	кластеризация, быстрый *retrieval*
Классификация	*Precision/Recall/F1, PR-AUC*	парафраз/верификация
Корреляция	*Pearson, Spearman*	*STS* по шкале
Ранжирование	*nDCG@k, MRR, Recall@k*	семантический поиск
Длинные тексты	покрытие сегментов, *max-avg*	документы/антиплагиат

Если модель предсказывает значение по шкале (например, 0–5), качество чаще измеряют корреляцией:

1 *Pearson* измеряет линейную связь между предсказанием и эталоном: хороша, когда важны именно численные значения, а не только порядок.

2 *Spearman* измеряет корреляцию рангов: полезна, когда важнее, чтобы модель правильно упорядочивала пары по близости (устойчива к нелинейным преобразованиям шкалы).

На практике корректно публиковать обе метрики: *Pearson* отражает «калибровку по шкале», *Spearman* – «согласованность порядка».

В семантическом поиске модель формирует ранжированный список документов. Здесь критичны ранговые метрики:

1 *Recall@k* – доля запросов, для которых релевантный документ попал в *top-k*. В двухэтапных системах это ключевой показатель первого этапа *retrieval*: если релевантный документ не найден, *re-ranking* не поможет.

2 *MRR* – среднее значение 1/*rank* первого релевантного документа; особенно важно для *QA*-сценариев, где нужен один лучший результат.

3 *nDCG@k* – учитывает позиции и, при необходимости, градуированную релевантность; обычно считается наиболее информативной метрикой качества выдачи на верхних позициях.

При сравнении документов (например, отчётов, курсовых, диалогов) «средний» эмбеддинг документа может скрывать локальные совпадения: часть текста может быть близкой, а часть – полностью оригинальной. Поэтому применяют сегментацию на предложения/абзацы и оценивают:

1 *max-avg similarity* – для каждого сегмента документа найти наиболее похожий сегмент в базе и усреднить;

2 *coverage*/покрытие – доля сегментов, для которых найден близкий сегмент выше порога (часто используется как компонент коэффициента заимствования);

3 Двухэтапную схему – быстрый поиск кандидатов по сегментам (контроль *Recall@k*), затем точная проверка *cross-encoder*'ом или более строгими правилами.

Демонстрационный протокол включает три типа задач:

1 *STS*-регрессия – сравнение предсказаний со шкалой человеческих оценок (*STS-B*, *SICK*).

2 Бинарная классификация – определение парафразов (*MRPC*) с подбором порога по *F*1 на валидации.

3 *Retrieval*-ранжирование – оценка качества выдачи по *nDCG@k* и *MRR* на корпусе с размеченной релевантностью (например, *BEIR* или доменный индекс).

Ниже приведен минимальный пример расчёта *Pearson*/*Spearman* для *STS-B* и *F*1 с подбором порога для *MRPC*. Для *retrieval*-задач расчёт *nDCG@k* и *MRR* выполняют по ранжированным спискам, полученным через векторный индекс (например, *FAISS*).

```
import numpy as np
from datasets import load_dataset
from sentence_transformers import SentenceTransformer
from scipy.stats import pearsonr, spearmanr
from sklearn.metrics import f1_score, precision_recall_curve
m = SentenceTransformer(«sentence-transformers/all-mpnet-base-v2»)
ds = load_dataset(«stsb_multi_mt», «en», split=«validation»)
e1 = m.encode(ds[«sentence1»], normalize_embeddings=True)
e2 = m.encode(ds[«sentence2»], normalize_embeddings=True)
pred = (e1 * e2).sum(axis=1)
gold = np.array(ds[«similarity_score»]) / 5.0
print(pearsonr(pred, gold).statistic, spearmanr(pred, gold).statistic)
mrpc = load_dataset(«glue», «mrpc», split=«validation»)
a = m.encode(mrpc[«sentence1»], normalize_embeddings=True)
b = m.encode(mrpc[«sentence2»], normalize_embeddings=True)
score = (a * b).sum(axis=1)
prec, rec, thr = precision_recall_curve(mrpc[«label»], score)
f1 = 2 * prec * rec / (prec + rec + 1e-9)
best_thr = thr[np.argmax(f1[:-1])]
```

```
    y_hat = (score >= best_thr).astype(int)
    print(«best_thr=«, float(best_thr), «F1=«, float(f1_score(mrpc[«label»],
y_hat)))
```

Предположим, что сравниваются две модели: базовая (например, *SBERT*-вариант) и дообученная на доменных парах. В *STS*-задаче дообучение может повысить *Spearman*, но снизить *Pearson*, если модель стала точнее упорядочивать пары, но хуже восстанавливать линейную шкалу. В бинарной задаче (*MRPC*) рост *F*1 при одновременном падении *Precision* может быть неприемлем, если возрастает доля ложных совпадений (таблица 2).

Для *retrieval*-компоненты рост *nDCG@*10 при одновременном падении *Recall@*100 означает улучшение ранжирования на верхних позициях, но ухудшение полноты кандидатов. Это сигнал о необходимости донастройки первого этапа *retrieval* или увеличения *k*.

Таблица 2 – Пример «карты» результатов (условные значения)

Метрика	***STS-B***	***SICK***
Cosine similarity	0,87	0,81
Pearson	0,89	0,84
Spearman	0,88	0,83
Accuracy (порог 0,7)	–	–
F1-score	–	–
MRR (retrieval)	0,91	–
nDCG@10 (retrieval)	0,93	–

Для проектов, где модели обновляются регулярно, целесообразно автоматизировать расчёт метрик и ведение «паспортов качества» (*model cards*): фиксировать данные, протокол и значения показателей. Это снижает риск регрессий и повышает воспроизводимость (таблица 3).

Таблица 3 – Примеры автоматизации контроля качества

Компонент	**Что автоматизируется**	**Результат**
Пайплайн метрик	Запуск расчета *Pearson/Spearman, F1, nDCG@k* на фиксированных наборах	Сопоставимость экспериментов и быстрый цикл сравнения.
Валидация на нескольких датасетах	Оценка на *STS-B, SICK, MRPC* и доменных данных	Выявление слабых мест и доменной чувствительности.
Визуализация	Гистограммы оценок, матрицы корреляций, ошибки по диапазонам	Диагностика выбросов и калибровки.
Мониторинг	Регулярный пересчет метрик на свежих данных	Раннее обнаружение деградации качества.

Заключение. Метрики оценки моделей семантического сравнения не могут быть сведены к одному числу, поскольку сами задачи различаются по формату выхода и по пользовательскому сценарию. Для *STS*-регрессии ключевыми являются корреляционные показатели (*Pearson*/*Spearman*), отражающие согласованность с человеческими оценками. Для бинарных постановок (парафраз/не парафраз) критичны метрики, устойчивые к дисбалансу, – *F*1, *Precision*/*Recall* и *PR-AUC*, а также корректный выбор и калибровка порога. Для *retrieval*-систем принципиальны ранговые метрики (*nDCG@k*, *MRR*, *MAP*) и показатели полноты (*Recall@k*), так как качество выдачи определяется тем, насколько быстро пользователь получает релевантный ответ.

Практическая рекомендация заключается в использовании пакета взаимодополняющих метрик, позволяющих контролировать как геометрию эмбеддингов, так и качество решений и ранжирования. Дополнительно важны воспроизводимость протокола, статистическая проверка значимости различий и мониторинг качества на доменных данных. Такой подход позволяет выбирать метрики осознанно, связывая их с рисками ошибок и ограничениями вычислительных ресурсов, и тем самым повышать надежность систем семантического сравнения в реальных приложениях.

Список литературы

1. *Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2020.*
2. *Thakur N. et al. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. 2021.*
3. *Gao L. et al. SimCSE: Simple Contrastive Learning of Sentence Embeddings. 2021.*
4. *Ni J. et al. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. 2022.*
5. *Izacard G., Grave E. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. 2021.*
6. *Thakur N., Reimers N., Daxenberger J., Gurevych I. Augmented SBERT / Retrieval & Re-ranking baselines and evaluation protocols. 2023–2026.*

UDC 004.451.25

ANALYSIS AND SELECTION OF METRICS FOR ASSESSING SEMANTIC COMPARISON MODELS: FROM BASIC INDICATORS TO RANKING METRICS

Krez K.S., Khlistunova K.V., Kuzmichkina D.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Schneiderov E.N. – Cand. of Sci., associate professor, associate professor of the department of ICSD, vice-Rector for Academic Affairs

Annotation. This article systematizes approaches to evaluating semantic textual similarity (STS) models and substantiates the choice of quality metrics depending on the problem statement and model architecture. Similarity metrics for vector representations, binary classification metrics, correlation criteria for scale-based STS, and ranking metrics for information retrieval are considered. It is shown that using a single metric (e.g., correlation or accuracy) can lead to incorrect conclusions when transferring the model to applied scenarios, especially in two-stage retrieval + re-ranking pipelines. A practical experimental protocol is proposed, including threshold selection for validation, F1/PR-AUC calculation for verification, and nDCG@k/MRR/Recall@k for ranking tasks. An example of metric calculation on typical datasets is given, along with recommendations for interpreting the results, taking into account class imbalance, scale calibration, and search recall requirements.

Keywords: semantic textual similarity; STS; evaluation metrics; sentence embeddings; cosine similarity; Pearson correlation; Spearman correlation; F1-score; PR-AUC; ranking metrics; nDCG@k; MRR; Recall@k; bi-encoder; cross-encoder; semantic search.