特許請求項翻訳のためのLLMによる後修正とプロンプト最適化

Post-Editing and Prompt Optimization with Large Language Models for Patent Claim Translation

愛媛大学　大学院理工学研究科
枝松　泰志
2026 年愛媛大学卒業。

愛媛大学　大学院理工学研究科　教授
後藤　功雄
2014 年京都大学大学院情報学研究科博士後期課程修了。博士（情報学）。日本放送協会、株式会社国際電気通信基礎技術研究所、独立行政法人情報通信研究機構を経て 2024 年より現職。

愛媛大学　大学院理工学研究科　教授
二宮　崇
2001 年東京大学大学院理学系研究科情報科学専攻博士課程修了。博士 (理学)。2017 年より愛媛大学大学院理工学研究科教授。自然言語処理の研究に従事。

1 はじめに

特許文書の翻訳は発明の権利範囲を言語の壁を超えて正確に伝達し、国際的な審査・権利行使・訴訟において法的効力を確保するために不可欠である。そのため特許文書の翻訳が人手で行われているが、経済的・時間的なコストが大きく、コスト削減のために、翻訳の自動化を目的とした特許文書に特化した機械翻訳の手法 [1] が提案されている。

特許文書のうち、特許請求項とは法的な保護範囲を定義する部分である。特許請求項は特許の中で法的な影響力を持ち、翻訳において高い正確性が求められる。特許請求項翻訳の課題として、一文が 100 文字を超えるほど長く、構成が複雑であるため正確な法的解釈を維持したまま翻訳することが困難であるということがある。さらに特許文書特有の表現や技術用語に対しての適切な訳語を選択することも困難である。近年、大規模言語モデル（LLM）の登場により、機械翻訳は既存の翻訳モデルを上回る性能を達成するようになった。特許請求項翻訳においても、LLM を用いて翻訳をした研究 [2] が行われており、高い性能を示している。しかし、LLM を用いた特許請求項翻訳の課題は依然として明確になっていない。

本稿は、WAT2025 において我々が提案した商用 LLM を用いた三つの翻訳手法 [3] について報告する。一つ目は Pinzhen Chen らの研究 [4] の手法に基づき、LLM による翻訳後に LLM-as-a-Judge[5] で翻訳品質を評価し、その結果を用いて翻訳を修正する後修正翻訳手法を提案する。二つ目は特許請求項に特化したプロンプトを用いて翻訳するプロンプト修正翻訳手法である。三つ目は実際の特許請求項を用いた Few-shot 学習に基づく翻訳手法である。また、本研究で提案する手法を用いて、WAT2025 の日英特許請求項翻訳タスクに参加した。これは日英両方向の特許請求項翻訳の精度を競いながら、正確な特許請求項翻訳の自動評価指標の確立を目指すワークショップである。今回は日英方向のタスクに参加した。

LLM-as-a-Judge を用いた評価実験の結果、LLM を用いた後修正翻訳手法で有効性が確認された。一方で、プロンプト修正翻訳手法や Few-shot 翻訳手法では効果は見られなかった。これに対し、LLM-as-a-Judge を用いた後修正翻訳のみを対象に実施された人手評価で

は改善が確認されなかった。この結果、本研究における LLM-as-a-Judge の性能が十分でないことが確認された。 また、自動評価指標では全ての手法において性能改善の効果が確認でき、特にプロンプト修正翻訳で高い性能改善の効果が確認できた。

2 LLM-as-a-Judge を用いた後修正

本手法では、特許請求項翻訳における課題である長文かつ複雑な構造を克服するため、LLM を用いて、人間の翻訳作業で一般に行われる工程を再現する。具体的には初回翻訳、翻訳結果の評価（LLM-as-a-Judge）、および評価に基づく修正(後修正翻訳)という三つのステップを再現する。

まず、初回翻訳として、原文を LLM に入力して翻訳する。入力する原文は元の改行情報を保持したまま請求項単位で与える。これは請求項における改行情報が文構造や係り受け関係を反映することが多く、LLM が文構造を正しく理解するための情報として有用と考えたためである。図 1 に、モデルに入力したプロンプト例を示す。

初回翻訳の生成後、原文および翻訳文を入力として、参照文を用いずに LLM-as-a-Judge による翻訳品質評価を実施する。評価は表 1 の独自の六つの観点に基づいて行い、各項目についてエラー内容の記述と 100 点満点による評価点数を出力させる。図 2 に、モデルに入力したプロンプト例を示す。

LLM-as-a-Judge による評価の後、出力された評価結果、原文、および初回翻訳を入力として LLM に与え、最小限の修正のみを施した英文を生成させる。図 3 に、モデルに入力したプロンプト例を示す。

初回翻訳プロンプト

```
あなたは特許請求項の日本語→英語 翻訳の専門翻訳者です.
以下の完全なポリシーを厳格に守ってください.
与えられた単一の日本語行を，米国特許クレーム風の英語として「必ず 1 行」に翻訳してください.
返すのは翻訳結果のみ（注釈なし）. 内容の追加・省略・分割・結合を禁止します.
出力は最終的な英語クレーム行のみ. ラベル，注釈，Markdown は禁止.
入力 1 行→出力 1 行を厳守（余計な改行を入れない）.
必ず 1 文のみ. 末尾はピリオドで終える.
法的スコープを完全に保持. 内容の追加・省略・分割・結合・再解釈は禁止.
クレーム種別（装置/システム/方法/組成/用途/プログラム/媒体），番号，他クレーム参照（従属関係）を保持.
数値・単位・記号・式・不等号・ラベル・範囲は全て厳密に保持（ただし後述の ASCII 正規化は例外）.
ASCII のみで出力. 非 ASCII 文字は ASCII 等価に置換（ASCII ルール参照）. “”^^e2^^80^^93 — ～ μ ° ” などは絶対に出さない.
従属クレームや wherein で新しい要素/限定を勝手に導入しない.
曖昧さを避ける."and/or" は絶対に使わない.
この単一の日本語クレーム行を英語に翻訳してください:
\{ここに実際の日本語文が入る.\}
```

図 1　初回翻訳プロンプト

表 1　LLM-as-a-Judge の指標

指 標
法的範囲の忠実性
米国特許クレームの様式，構造の適合
数値・単位・範囲・式の正確性
先行詞対応・参照整合性
用語の正確さ・統一性
表現の自然さ・読みやすさ

LLM-as-a-Judge プロンプト

```
あなたは非常に几帳面な特許請求項レビュアです\\
参照訳（正解訳）は一切使わず，原文日本語（JA）に対して英訳（PRED）を評価してください.\\
次の評価軸と出力形式に従ってください.\\

評価カテゴリ（各 0^^e2^^80^^93100，6 カテゴリは等加重）:\\
- fidelity\_legal\_scope（法的スコープの忠実性）\\
- us\_style\_structure（米国クレーム文体・構造）\\
- numbers\_units\_ranges（数値・単位・範囲）\\
- antecedent\_dependency（先行詞・従属関係）\\
- terminology（用語）\\
- naturalness（自然さ）\\

出力に以下のセクションを含めてください:

### Findings\\
- 具体的な問題点，または問題ない点（確認）を箇条書き.\\
  （法的文体，忠実性，数値/単位/式/範囲，用語一貫性，先行詞，従属関係，句読点/フォーマット，自然さ等）
- 各項目は [Fidelity]，[Numbers/Units] などのラベルで開始し，JA/PRED の該当箇所を正確に引用してください.

### Fix Suggestions\\
- Findings に対応づけた箇条書きで，最小限の修正で済む「修正案（英語断片）」を示してください.
```

図 2　LLM-as-a-Judge プロンプト

後修正翻訳プロンプト

```
あなたは特許請求項のプロ翻訳者で，第二稿（2nd pass）を作成します.\\
以下のポリシーを厳守してください. 参照訳は一切見ません.\\
与えられる「日本語行」「初回英訳」「評価者によって評価された評価」をもとに，\\
内容を追加・削除せずに英訳を修正してください.\\
出力は米国クレーム文体の英語 1 行のみ.\\


### Japanese（与えられたスコープを超えて翻訳しない）\\
\{実際の原文\}\\

### First-pass English\\
\{初回翻訳結果\}\\

### Issues to fix（抽象的. 参照文は使っていない／与えられていない）\\
\{LLM-as-a-Judge の結果\}\\

上記を踏まえて英語行を書き直してください（1 行のみ）.
```

図 3　後修正翻訳プロンプト

3 特許請求項に特化したプロンプト修正

本手法では 2 節で導入した翻訳プロンプトを改良し、意味と法的効力を保持しつつ、米国特許請求項の慣習に、より忠実に従った翻訳を目指す。具体的には米国特許スタイルで翻訳すること、先行する他の請求項を引用しない独立請求項と引用する従属請求項とを明確にすること、句読点の統一、冠詞の用法などを正確に行うことを明記した。2 節のプロンプトに、図 4 のプロンプトを追加した。

```
プロンプト修正翻訳用プロンプト
原則として開放的遷移語は "comprising:"\\
並列要素の区切りはセミコロン.\\
2 個以上の並列列挙では最後の要素の前に "; and" を入れる.\\
要素名は明確にし，同じ要素は同じ語で一貫して参照する.\\
先行する他の請求項を引用しない独立請求項と引用する従属請求項とを明確にする.\\
  *既出要素への限定のみ追加. 新しい要素は導入しない.* \\
初出: "a/an/at least one [X]"（または "a first [X]", "a second [X]"）\\
- 再出: "the [X]"\\
- 定義後に "a/an" に戻さない. 単数/複数も一貫.\\
  - 各 wherein 節を，正しい先行要素に結び付ける.\\
- 並列条件は構造化して書く:\\
  "wherein: (i) ...; and (ii) ... ."\\
- wherein 節で新しい要素は導入しない.\\
```

図 4　プロンプト修正翻訳用プロンプト

```
検索用プロンプト
"あなたは日本語特許請求項の専門家です.
\\以下の日本語 1 行（1 クレーム）について，"\\
"FAISS で高精度に用例を拾うための『日本語クエリ』を**ちょうど 3 件**，JSON 配列で
出力してください."\\
"一般語（装置，処理，データ等）や曖昧語を避け，品詞は名詞句中心で**8〜24 文字程
度**に収めます."\\
"括弧・全角記号・機種依存文字は使用しません."\\
"説明や余計な文字は一切付けないでください."\\
```

図 5　検索用プロンプト

4 特許請求項を用いた Few-shot 学習に基づく翻訳

本手法では特許請求項翻訳における専門用語や特許特有の表現を正確に翻訳するために、FAISS[6] および SentenceTransformer[7] により検索された翻訳例を用いた Few-shot 翻訳を行った。FAISS は大規模なベクトルの集合から類似したベクトルを効率よく検索することを目的とした高速なベクトル類似度検索ライブラリである。FAISS インデックスの構築にあたっては、日英特許出願コーパスである JaParaPat[8] を用いた。本コーパスは日本語および英語の特許文書間でアライメントされた大規模な日英対訳コーパスである。2016 年から 2020 年の間に出願された特許文書ファミリーから自動抽出された、約 1 億文対から構成されており、出願種別ラベルや文書 ID などのメタデータを含んでいる。本研究では、出願種別ラベルのうち PCT ルートに分類されている日英対訳文対を使用する。PCT ルートでは単一の国際特許出願が翻訳を通じて複数の国の特許庁に提出されるため、その結果得られる多言語公開文書は実質的に対訳関係にある。これらの文書は同一出願に対する直接的な翻訳であることから、高い信頼性を有する対訳データとみなすことができる。また、本研究は、請求項の文のみを使用した。日本語の請求項文は SentenceTransformer に基づく、多言語埋め込みモデル（intfloat/multilingual-e5-base）[9] を用いてベクトルに埋め込み、文間の意味的類似度をコサイン類似度に基づいて評価できるようにする。これにより、入力された請求項に対して意味的に類似した日英文対を FAISS によって検索し、得られた用例を例文としてプロンプトに加えて Few-shot 翻訳を行うことが可能となる。

本研究では二種類の Few-shot 手法を用いた。一つ目の手法では特許請求項一文を FAISS インデックスに対して検索にかけ、類似度の高い上位三つの日英文対を取得し、それらを例文として翻訳プロンプトに挿入する。二つ目の手法では原文中に重要な用語を含む翻訳例を取得するため、まず LLM を用いて入力文から三つの用語を抽出する。図 5 にモデルに入力したプロンプトを示す。こうして抽出した用語を FAISS インデックスで検索にかけ、各用語との類似度が最も高い日英文対を取得し、合計三つの日英文対を例文として翻訳プロンプトに挿入する。

表 2　各データの請求項数

	検証データ	評価用データ
特許数	13	26
特許請求項数	19	70

5 実験

5.1　実験設定

本研究では基盤となる LLM として OpenAI の GPT-5[10] を使用する。

本研究では WAT2025 の「Patent Claims Translation/Evaluation Tasks」において提供されている公式の検証、評価データを使用した。検証データは原言語の特許請求項とそれに対応する翻訳文から構成されているのに対し、評価データでは原言語の特許請求項のみが含まれている。これらのデータは特許ごとにファイルが用意されており、その中に特許請求項が存在している。各データセットに含まれている特許数および特許請求項数を表 2 に示す。

特許請求項は非常に長く、構造的に複雑な文を含むため、各請求項の構造を完全に理解することが難しい。さらに翻訳の正確性は意味だけでなく、法的範囲や技術用語の適切性といった多次元的観点から維持される必要がある。そこで本研究では、LLM を用いて翻訳品質を評価する LLM-as-a-Judge を採用し、長文全体にわたる翻訳の適切性を一貫して評価する。また WAT2025 における公式評価として、タスク主催者による人手評価が実施された。本研究では、これら二つの評価指標を用いて特許請求項翻訳の品質を評価すると共に、LLM-as-a-Judge による評価の妥当性および LLM を用いた特許

請求項翻訳の課題について調査する。

また、本研究の主たる分析は LLM-as-a-Judge および人手評価に基づくが、特許請求項翻訳を従来の自動評価指標がどのように評価するかを確認することも有益であると考え、補足的な評価として既存の自動評価も実施した。

人手評価では各原文およびそれらに対応する翻訳文に対して、手動によるエラー注釈および評価スコアが付与された。エラー注釈は誤訳、訳抜け、湧き出しなどのエラーカテゴリに基づいて問題のある箇所に付与され、各エラーに重大度（Major または Minor）がラベル付けされた。 さらに各文に対して 100 点満点のスコアが割り当てられた。翻訳結果に評価優先度を付与し、LLM-as-a-Judge を用いた後修正翻訳のうち、初回翻訳と後修正翻訳の二つの翻訳結果がタスク主催者によって評価された。各翻訳結果に対して、70 文からなる評価文のうち 28 文が人手により評価された。

LLM-as-a-Judge による評価では、翻訳文を LLM が評価する LLM-as-a-Judge を用いる。評価基準は表 1 で定義したものと同一のものである。

自動評価指標として COMET[11] および BLEU[12] を用いた。評価データは参照訳がなく、参照訳を使う自動評価手法は適用できないため、この自動評価では参照訳を含む検証データを用いて算出する。

5.2　人手評価の結果

人手評価における誤りカテゴリと評価スコアをそれぞれ表 3、表 4 に示す。初回翻訳と比較すると、後修正翻訳の出力は Major エラー数が 18 から 42 に増加し、Minor エラー数は 119 から 150 に増加した。その結果、総エラー数は 137 から 192 に増加し、平均評価スコアは 86.1 から 81.6 に低下した。

表 3　人手評価による誤りカテゴリ分類結果

	初回翻訳		後修正翻訳	
エラー種類	Major	Minor	Major	Minor
訳抜け	8	13	24	14
用語の一貫性	1	33	0	36
文法	1	8	2	1
誤訳	5	18	7	25
その他	0	2	0	5
文脈上不適切	3	11	2	14
湧き出し	0	10	5	34
原文エラー	0	2	1	2
句読点エラー	0	17	0	8
一貫性の欠如	0	0	0	4
ぎこちない	0	4	0	5
冠詞	0	0	1	2
エラー総数	18	119	42	150
エラー総数 (Major+Minor)	137		192	

表 4　人手評価による評価スコア

	初回翻訳	後修正翻訳
平均スコア	86.1	81.6

5.3　LLM-as-a-Judge の結果

LLM-as-a-Judge による評価結果を表 5 に示す。LLM による後修正翻訳では、初回翻訳よりも高いスコアを達成した。一方で、プロンプト修正翻訳のスコアは初回翻訳と比較して低下し、Few-shot（全文検索）および Few-shot（用語検索）のいずれも初回翻訳より低いスコアを示した。したがって、LLM-as-a-Judge の評価では、これらの Few-shot およびプロンプト修正翻訳の手法の有効性は確認されなかった。

表 5　LLM スコア

システム名	LLM-as-a-Judge スコア (%)
初回翻訳	91
後修正翻訳	92
プロンプト修正翻訳	80
Few-shot（全文検索）	84
Few-shot（用語検索）	78

5.4　自動評価

実験結果を表 6 に示す。初回翻訳と比較して、提案手法全てにおいて COMET は上昇し、プロンプト修正翻訳と Few-shot 翻訳では BLEU も上昇した。

表 6　自動評価

システム名	COMET(%)	BLEU(%)
初回翻訳	84.6	53.8
後修正翻訳	84.9	48.9
プロンプト修正翻訳	85.4	56.6
Few-shot（全文検索）	85.1	53.9
Few-shot（用語検索）	85.0	53.5

5.5　人手評価が実施された翻訳の分析

表 3 の人手評価の結果を元に特許請求項翻訳の分析を行う。文法誤りやカンマ・セミコロンの使用といった句読点に関する表層的な誤りは改善されたものの、他の種類の誤りについては改善が見られなかった。特に湧き出し、訳抜け、誤訳が大幅に増加しており、原文の特許請求項への忠実性に関わる誤りが増加したことが確認され、これらの数が多いほど翻訳スコアは減少した。 具体的には「～手段」の「手段」や「～量」の「量」などの訳抜けや、「each of」や「configured to」などの湧き出しといった、細かい単語レベルのエラーがよく見られた。よく見られた誤訳のエラーは「a」と「the」の誤選択や脱落といった冠詞のエラーに起因していた。これらの冠詞のエラーを除くと、誤訳のエラーでは初回翻訳は 11 個、後修正翻訳は 13 個で差は小さかった。また、湧き出しのエラーとして、原文では存在しなかったが要素を並列するときの「(i)」、「(ii)」の記号が湧き出しとしてカウントされた。一方で用語の一貫性や文脈上の不適切さに関する誤りも多数観察された。

今回、LLM-as-a-Judge の評価指標では法的な忠実性の他に米国特許クレームの様式や表現の自然さ、読みやすさといった翻訳の表層的な部分を同じ比重で評価している。その結果文法や句読点の修正、米国式請求項スタイルへの適合、特許翻訳で一般的な表現の導入といった表層的改善は見られたが、一方で範囲や比較方向の誤り、先行詞参照の誤り（誤訳）、原文に存在しない要素の追加（湧き出し）、および必須要素の脱落（訳抜け）などの原文忠実性の低下に関わる誤りが増加したと考えられる。特に、後修正翻訳では最小限の修正を行うのではなく、文全体を書き換える傾向が見られたため、原文への厳密な忠実性よりも整った英語表現の生成が優先された可能性がある。

人手評価と LLM-as-a-Judge 評価を比較すると、後修正翻訳では LLM-as-a-Judge の枠組みではスコアが向上した一方で、人手評価では翻訳品質が低下していた。したがって、本研究においては LLM-as-a-Judge フレームワークの性能が十分でなかったことが確認された。

5.6　人手評価が実施されていない翻訳の分析

プロンプト修正翻訳および Few-shot 翻訳について、LLM-as-a-Judge による評価結果を表 3 の基準を参考に分析した。プロンプト修正翻訳では米国特許請求項スタイルとの整合性、英語出力の自然さ、用語および単位表現の安定性を向上させた。一方で、請求項の権利範囲に関する不正確な修正や、誤った従属関係の挿入により、法的範囲の忠実性や先行詞対応・参照整合性のスコアは低下した。これはモデルが「自然な英語」の生成および「米国式請求項スタイル」への準拠に強く偏る傾向があり、その結果、構造保持や法的忠実性といった特許翻訳において本質的な側面が損なわれやすいためである。この忠実性の低下が、LLM-as-a-Judge の評価スコアの低下につながったと考えられる。

同様に、Few-Shot 翻訳では、句読点の配置、要素列挙、「configured to」の使用に代表される語彙的一貫性、および米国特許請求項特有の文構造の安定化など、表層的改善が見られた。しかし、Few-Shot 翻訳は文体的一貫性や用語の安定性を向上させる一方で、検索された例文の構造的バイアスの影響を強く受けるため、要素の再編成、節位置の移動、限定条件を導入するためによく使われる wherein 句の不要な挿入といった構造的歪みが翻訳出力に生じ、LLM-as-a-Judge の評価スコアの低下につながったと考えられる。以上の分析から、プロンプトによって LLM に制約を課した場合、本研究で用いたモデルでは、全体的な内容の忠実性を低下させることなく表層的な制約を満たすことが難しいことが示唆された。これは、今後の研究における重要な課題である。

6　まとめ

特許請求項の翻訳に焦点を当てて、LLM を用いた特許請求項翻訳の品質向上と課題分析を行った。本研究では LLM-as-a-Judge を用いた後修正翻訳、プロンプト修正翻訳、Few-shot 翻訳の三つの異なるアプローチについて提案と比較を行った。その結果、LLM-as-a-Judge を

用いた後修正翻訳では LLM-as-a-Judge による評価スコアは改善した。一方で、プロンプト修正翻訳と Few-shot 翻訳は、LLM-as-a-Judge 評価において改善を示さなかった。これに対して、公式の人手評価では、LLM-as-a-Judge を用いた後修正翻訳において、初回翻訳と後修正翻訳を比較した結果、総エラー数は増加し、平均スコアは初回翻訳の 86.1 から後修正翻訳の 81.6 に低下しており、人手評価に基づく品質が悪化した。この結果は、本研究において LLM-as-a-Judge フレームワークの性能が十分ではなかったことを示している。

分析の結果、提案手法の後修正翻訳は初回翻訳に比べて表層的な品質エラーの改善には寄与したものの、訳抜け、湧き出しといった日英文対での対応関係の誤りや係り受け関係の誤りといった元の特許請求項に対する忠実性に関してエラーが増加した。プロンプトによって LLM に制約を課した場合、本研究で使用したモデルでは、全体の忠実性を低下させることなくそれらの制約を満たすことが難しいという結果になった。

今後の課題として、まず日英文対における語単位での対応関係を厳密に保持し、元の特許請求項に対する原文忠実性の高い翻訳を行うことが挙げられる。また特許請求項翻訳に適した LLM-as-a-Judge を確立し、人手評価指標との相関を実証することも今後の課題である。

参考文献

[1]Zi Long, Takehito Utsuro, Tomoharu Mitsuhashi, Mikio Yamamoto. Translation of Patent Sentences with a Large Vocabulary of Technical Terms Using Neural Machine Translation, 2017.

[2]Haruto Azami, Minato Kondo, Takehito Utsuro, Masaaki Nagata. Patent Claim Translation via Continual Pre-training of Large Language Models with Parallel Data. In Proceedings of Machine Translation Summit XX: Volume 1, pp. 300–314, 2025.

[3]Taishi Edamatsu, Isao Goto, Takashi Ninomiya. Ehime-U System with Judge and Refinement, Specialized Prompting, and Few-shot for the Patent Claim Translation Task at WAT 2025. In Proceedings of The 12th Workshop on Asian Translation (WAT 2025), 2025.

[4]Pinzhen Chen, Zhicheng Guo, Barry Haddow, Kenneth Heafield. Iterative Translation Refinement with Large Language Models. In Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1), pp. 181–190, 2024.

[5]Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, Jian Guo. A Survey on LLM-as-a-Judge, 2025.

[6]Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, Hervé Jégou. The Faiss library. arXiv 2401.08281, 2024.

[7]Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019.

[8]Masaaki Nagata, Makoto Morishita, Katsuki Chousa, Norihito Yasuda. JaParaPat: A Large-Scale Japanese-English Parallel Patent Application Corpus. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, pp. 9452–9462, 2024.

[9]Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, Furu Wei. Multilingual E5 Text Embeddings: A Technical Report, 2024.

[10]OpenAI. OpenAI GPT-5 System Card, arXiv 2601.03267, 2025.

[11]Ricardo Rei, Craig Stewart, Ana C Farinha, Alon Lavie. COMET: A Neural Framework for MT Evaluation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 2685–2702,

2020.
[12]Matt Post. A Call for Clarity in Reporting BLEU Scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191, 2018.