DiBiMT: A Gold Evaluation Benchmark for Studying Lexical Ambiguity in Machine Translation

Federico Martelli[1*], Stefano Perrella[1*], Niccolò Campolungo[2*], Tina Munda[3], Svetla Koeva[4], Carole Tiberius[5], and Roberto Navigli[1*]

[1]Sapienza University of Rome, Department of Computer, Control, and Management Engineering
martelli@diag.uniroma1.it, perrella@diag.uniroma1.it, navigli@diag.uniroma1.it

[2]Litus AI
nick@litus.ai

[3]University of Ljubljana, Centre for Language Resources and Technologies
tina.munda@cjvt.si

[4]Institute for Bulgarian Language, Bulgarian Academy of Sciences
svetla@dcl.bas.bg

[5]Dutch Language Institute/Leiden University Centre for Linguistics
carole.tiberius@ivdnt.org

Despite the remarkable progress made in the field of Machine Translation (MT), current systems still struggle when translating ambiguous words, especially when these express infrequent meanings. In order to investigate and analyze the impact of lexical ambiguity on automatic translations, several tasks and evaluation benchmarks have been proposed over the course of the last few years. However, work in this research direction suffers from critical shortcomings. Indeed, existing evaluation datasets are not entirely manually curated, which significantly compromises their reliability. Furthermore, current literature fails to provide detailed insights into the nature of the errors produced by models translating ambiguous words, lacking a thorough manual analysis across languages.

*With a view to overcoming these limitations, we propose Disambiguation Biases in MT (*DiBiMT*), an entirely manually curated evaluation benchmark for investigating disambiguation biases in eight language combinations and assessing the ability of both commercial and non-commercial systems to handle ambiguous words. We also examine and detail the errors produced by models in this scenario by carrying out a manual error analysis in all language pairs.*

* Equal contribution.

Action Editor: Anh Tuan Luu. Submission received: 31 January 2024; revised version received: 6 July 2024; accepted for publication: 27 August 2024.

https://doi.org/10.1162/coli_a_00541

Additionally, we perform an extensive array of experiments aimed at studying the behavior of models when dealing with ambiguous words. Finally, we show the ineffectiveness of standard MT evaluation settings for assessing the disambiguation capabilities of systems and highlight the need for additional efforts in this research direction and ad-hoc testbeds such as DiBiMT. *Our benchmark is available at:* `https://nlp.uniroma1.it/dibimt/`.

1. Introduction

Over the course of the last few decades, the field of Machine Translation (MT) has witnessed remarkable advances in terms of fluency and idiomaticity of output translations as well as model efficiency, largely due to the development of the Transformer architecture and the attention mechanism (Vaswani et al. 2017). Recently, with the scaling of model size and the increased availability of data, Large Language Models (LLMs) are bringing a breeze of change to virtually all Natural Language Processing (NLP) tasks, by combining the extraordinary representational power of modern language modeling techniques with in-context learning (Dong et al. 2024). Specifically, in MT, LLMs have demonstrated remarkable translation capabilities in several language combinations (Brown et al. 2020; Zhu et al. 2024), despite weaker performance on less-represented languages (Kocmi et al. 2023).

Notwithstanding such advancements, current systems still struggle when dealing with specific linguistic phenomena, among which **lexical ambiguity** poses one of the greatest challenges (Emelin, Titov, and Sennrich 2020; Campolungo et al. 2022; Iyer et al. 2023). Considered one of the hardest problems in NLP, lexical ambiguity is a pervasive linguistic phenomenon in which a given word can express different meanings depending on the context in which it occurs (Krovetz and Croft 1992). Unlike other forms of linguistic ambiguity that can arise at different levels (e.g., the morphological, syntactic, or pragmatic ones), lexical ambiguity affects individual words or expressions and their meanings.

Dealing with this problem properly is of vital importance to ensure high-quality output translations. In order to illustrate the impact of lexical ambiguity on the translation process, let us consider the examples reported in Table 1, in which we show the automatic translations of two source sentences from English into four target languages, namely, German (DE), Italian (IT), Russian (RU), and Spanish (ES), obtained with two different systems: DeepL,[1] a state-of-the-art commercial MT system, and ChatGPT,[2] a popular chatbot based on GPT-3.5.[3] Both source sentences contain an ambiguous word highlighted in violet, namely, the noun ***agency*** and the verb ***bark***, respectively. We identify the corresponding translations of the ambiguous source words and mark them in green with a straight underline or red with a wavy underline, depending on whether they are to be considered correct or not. For instance, the first source sentence *An example is the best* ***agency*** *of instruction* is translated into Italian with *Un esempio è la migliore* ***agenzia*** *di istruzione*. In this context, the ambiguous word ***agency*** indicates how a specific goal is achieved, while the Italian word *agenzia* does not express the same meaning, resulting in an incorrect translation. In this case, correct choices would instead be *maniera*, *mezzo*, or *modo*. Similarly, the second source sentence ***Bark*** *the roof*

1 `https://www.deepl.com/translator`.
2 `https://openai.com/chatgpt/`.
3 The translations were generated in February 2023.

Table 1
Examples of correct and incorrect translations of two sentences in English, each containing an ambiguous word, namely the noun *agency* and the verb *bark*, respectively. The first input sentence was translated using DeepL, whereas the second input sentence was translated using ChatGPT (based on GPT-3.5).

Input	EN	An example is the best **agency** of instruction. **Bark** the roof of a hut.
Output	DE	Ein Beispiel ist das beste **Mittel** zur Belehrung. **Die Rinde** des Daches einer Hütte **abschälen**.
	ES	Un ejemplo es la mejor **agencia** de instrucción. **Pelar la corteza** del techo de una choza.
	IT	Un esempio è la migliore **agenzia** di istruzione. **Sbucciare la corteccia** del tetto di una capanna.
	RU	Примером может служить лучшее **агенцтво** по обучению. **Покрывать корой** крышу хижины.

of a hut could be translated as ***Rivestire*** *il tetto di una capanna*. Instead, this source text is incorrectly rendered with ***Sbucciare la corteccia*** *del tetto di una capanna*, in which the expression *sbucciare la corteccia* literally means *peel the bark*.

In order to produce high-quality translations, MT systems must be able to determine the correct meaning of ambiguous source words and translate them with the corresponding senses into a given target language (Marvin and Koehn 2018). When systems fail to select the correct sense and, instead, choose a wrong one, a disambiguation error is produced (see Table 1). According to recent research work, these errors can be related to artifacts located in the training data, which lead to **disambiguation biases**. These can be described as the tendency of systems to translate a given ambiguous word used with an infrequent meaning into a target language with an incorrect, often predominant, sense (Emelin, Titov, and Sennrich 2020; Campolungo et al. 2022). In this scenario, investigating the disambiguation capabilities of systems involves studying the quantity and nature of semantic errors produced and, consequently, detecting the presence of disambiguation biases. However, despite the work proposed so far, we still lack a comprehensive study of this phenomenon and its impact on automatic translations across different languages. To fill this gap, the present article aims to provide insightful answers to the following research questions:

- **RQ1** What is the impact of lexical ambiguity on state-of-the-art MT models and LLMs?
- **RQ2** What error patterns can be identified, when analyzing the errors produced by models in translating ambiguous words?
- **RQ3** Given an encoder-decoder architecture trained for MT, to what extent does the encoder contribute to distinguishing word senses? Does

the encoder learn representations suitable for disambiguating ambiguous words?

- **RQ4** What is the relation between the capacity of an architecture and its ability to represent different senses of ambiguous words?
- **RQ5** Do multilingual models sacrifice their disambiguation capabilities so as to be able to handle multiple languages?
- **RQ6** What is the impact of the beam search decoding strategy on the disambiguation capabilities of MT models?
- **RQ7** How effective are standard evaluation settings in assessing the disambiguation ability of MT systems?

This article extends Campolungo et al. (2022), who proposed a manually curated benchmark for investigating disambiguation biases in five language pairs. In our work, we address the aforementioned research questions by bringing about the following novel contributions:

- We extend the benchmark from five to eight languages: Bulgarian, Chinese, Dutch, German, Italian, Russian, Slovene, and Spanish. These languages cover four subgroupings of the Indo-European and Sino-Tibetan language families: Balto-Slavic, Germanic, Italic, and Sinitic.
- We significantly increase the coverage of the benchmark by manually expanding and refining the sets of good and bad translations.
- Compared with Campolungo et al. (2022) who tested only seven MT systems, we include four additional state-of-the-art MT systems and eight LLMs.
- We carry out a novel manual error analysis to identify and describe the error patterns shown by the tested systems when dealing with lexical ambiguity.
- We propose an extensive array of experiments:
  - (i) We investigate the impact of MT systems' encoder module on their disambiguation capabilities, thereby studying the effectiveness of the representations learned by the encoder for Word Sense Disambiguation (WSD).
  - (ii) We inspect the extent to which the capacity of a model is related to its WSD capabilities, and whether multilingual MT systems sacrifice their ability to disambiguate rarer senses in order to be able to deal with multiple languages.
  - (iii) We study the role of the beam search algorithm in dealing with lexical ambiguity, and whether systems are inherently biased toward (more common) senses of ambiguous words.
  - (iv) We explore the suitability of standard MT evaluation settings for detecting disambiguation errors, thus highlighting the need for a manually curated benchmark such as DiBiMT.

2. Related Work

One of the first contributions aimed at assessing the ability of MT models to handle lexical ambiguity is proposed by Mihalcea, Sinha, and McCarthy (2010), who introduce the Cross-Lingual Lexical Substitution (CLLS) task. Inspired by McCarthy and Navigli (2007), who put forward the English Lexical Substitution task at SemEval-2007, CLLS requires systems to provide a substitute for a focus word in context in a given target language. For example, a candidate system for English-to-Italian cross-lingual lexical substitution provided with the sentence *He removed the* ***top*** *of the carton* would be expected to suggest the substitutes *coperchio* and *parte superiore* for the focus word *top*. Instead, if the system is provided with the sentence *Put a* ***top*** *on the toothpaste tube*, such system would be expected to propose the word *tappo* as a substitute for *top*. Although the focus word is the same in both sentences, the two substitutes differ. Crucially, CLLS does not rely on any sense inventory, which presents a significant limitation when it comes to assessing the disambiguation capabilities of MT models. In fact, since focus words, which can be ambiguous, are not associated with a semantic tag, it is not possible to determine their meaning. As a result, the nature of potential semantic errors and biases cannot be investigated effectively. Furthermore, the task focuses solely on one language pair, namely, English-Spanish, and provides only a development set composed of 30 words and a test set of 100 words.

The aforementioned limitations are partially addressed by Lefever and Hoste (2010, 2013) who propose the Cross-Lingual Word Sense Disambiguation (CLWSD) task. Here, systems are required to choose the most appropriate translations from a set of possible candidates, for a given word in context. For instance, a system provided with the sentence *He removed the* ***top*** *of the carton* and candidate translations for *top*, that is, *coperchio*, *tappo*, and *parte superiore*, among others, should select only *coperchio* and *parte superiore* as correct translations. This approach addresses the well-known drawbacks of traditional WSD (Navigli 2009; Bevilacqua et al. 2021), such as the fine granularity and static nature of sense distinctions, and even the need for a wide coverage sense inventory. However, while some characteristics of the aforementioned task formulation are advantageous from a WSD standpoint, these might prove detrimental to the detection of semantic biases in MT. In fact, similarly to the CLWSD task, the lack of wide-coverage sense inventories hampers the study of the nature of disambiguation biases, making it impossible to determine whether a given incorrect sense chosen by an MT model is more frequent than the corresponding correct one. Moreover, the test dataset focuses on 20 ambiguous focus words, covering only one part of speech, namely, nouns. Finally, CLWSD relies on an automatically built sense inventory to select translation candidates. This inventory is derived from Europarl (Koehn 2005), which mostly covers the political domain; the test sentences are drawn from the American National Corpus (Ide and Suderman 2004), which, instead, encompasses different topics, thus introducing potential coverage issues in the sense inventory due to the topic mismatch.

Focused on investigating the disambiguation capabilities of MT systems, Gonzales, Mascarell, and Sennrich (2017, ContraWSD) propose a new contrastive evaluation dataset, in which every instance is composed of three elements: (i) a source sentence s containing a focus word; (ii) its reference translation t, and (iii) a set of contrastive examples $C = \{c_1, \ldots, c_n\}$, where the translation for the focus word is replaced with the translation of one of its other meanings. For each instance, a candidate MT system is required to assign a probability score to the pairs composed of s and t as well as s and all other contrastive sentences in C. Given a function ρ which assigns a probability score to a given (source, translation) pair, an instance is considered correctly classified if

$\rho(t|s) > \rho(c_i|s)\ \forall c_i \in C$, in other words, the pair containing the reference translation is assigned a higher score. ContraWSD includes 7,200 instances for German → English, and 6,700 for German → French. While the contrastive formulation simplifies the process of evaluating models by requiring only a scoring function, such a formulation exhibits a number of significant drawbacks. First, this evaluation disregards the fact that a given MT system might never generate either the translation labeled as correct or the contrastive ones, which exposes the evaluation to the degree to which a given translation is in-distribution compared to the training data. Moreover, while the authors enforce that the correct and incorrect translations agree in number and gender (depending on the target language), models could assign lower scores due to poor fluency of the artificially created sentences, rather than actual semantic errors, thanks to the strong language modeling capabilities exhibited by MT models (Voita, Sennrich, and Titov 2021).

Building upon this work, Rios, Müller, and Sennrich (2018) present the Word Sense Disambiguation Test Suite, that is, a denoised version of ContraWSD focused on the German → English language combination. Here, rather than scoring translations, the WSD Test Suite evaluates the MT output directly. Importantly, this new dataset introduces a crucial constraint that enables straightforward disambiguation of the sense chosen by candidate models, by including only sentences in which the translation of the ambiguous source word cannot refer to different meanings. In order to illustrate this constraint, let us consider the example provided by the authors: The German ambiguous word *Stelle* can refer to two different senses, namely, *job* and *place*, intended as *a paid position of regular employment* and as *a particular position, point, or area in space; a location*, respectively; however, since both of these senses could be translated into English with *position*, the authors remove the word *Stelle* from the dataset. WSD Test Suite represents a significant leap toward a reliable evaluation of semantic biases in automatic translations, with 3,249 sentence pairs, each targeting one of the 20 ambiguous German words in the dataset, and totaling 45 distinct word senses. Despite the aforementioned improvements, the WSD Test Suite displays limited coverage in terms of words and senses, and features only one translation direction, namely from English to German, while still exhibiting the significant drawbacks that the contrastive formulation suffers from, as illustrated above.

The first effort to create a large-scale contrastive dataset for detecting disambiguation biases is made by Raganato, Scherrer, and Tiedemann (2019), who propose the Multilingual Contrastive WSD benchmark (MuCoW). Inspired by both ContraWSD and the WSD Test Suite, MuCoW is an automatically created test suite available in two variants, namely, the scoring and the translation variant. While the former contains more than 200,000 sentence pairs derived from word-aligned parallel corpora in 16 translation directions, the latter includes 9 language pairs with a total of 15,600 sentences. The MuCoW dataset is created in three steps. First, an alignment tool is used to obtain aligned word pairs occurring more than 10 times in parallel corpora from the OPUS collection (Tiedemann 2012), each connected to at least two distinct focus words. Second, the authors cluster the translations by relying on BabelNet (Navigli and Ponzetto 2010; Navigli et al. 2021) and sense embedding similarity, so as to address BabelNet's fine granularity. Third, random sentences are selected for a given word pair and the target word is replaced with a lexicalization pertaining to a synset other than that associated with the focus ambiguous word, thus constructing several contrastive instances for a given pair (scoring variant only). Despite the aforementioned advantages, we highlight two key limitations in MuCoW over and beyond the contrastive formulation itself, whose drawbacks have been illustrated above. First, due to its entirely automatic construction, MuCoW is prone to contain errors and

noise derived from the underlying semi-automatic sense inventory. Second, such a benchmark relies on the problematic assumption that the synonyms associated with a specific synset can be used interchangeably for translating a given focus word.

Moving in a different direction from previous works, Emelin, Titov, and Sennrich (2020) take a step toward model analysis, thoroughly exploring the correlation between biases picked up by MT models and the distribution of their training data. Interestingly, the authors leverage this correlation to create two challenge sets: (i) a WSD bias challenge set, built to quantify the intrinsic bias caused by an ambiguous word's context; and (ii) an adversarial challenge set, built to measure models' susceptibility to adversarial injection of terms usually associated with meanings other than the target one. Both challenge sets rely on manually refined sense clusters, initially built by automatically merging together BABELNET synsets. Each sense cluster contains an ambiguous English word and a set of monosemous German words, such that, considered jointly, they uniquely identify a specific meaning. Crucially, this work requires the availability of training data which are not always accessible. Furthermore, it focuses on just one language combination, namely, English → German. The approach relies solely on the accuracy score to evaluate models, and this can only provide partial information regarding the biases exhibited by models.

Based on the findings, drawbacks, and open research questions discussed in previous works, Campolungo et al. (2022) propose DIBIMT as a framework aimed at investigating not only the presence but also the nature and properties of semantic biases in MT in multiple language combinations, covering both nominal and verbal senses. In this article, we consolidate and extend the aforementioned framework and put forward a thorough study of the impact of lexical ambiguity on MT.

3. The DIBIMT Benchmark

We now introduce Disambiguation Biases in MT (DIBIMT), an entirely manually curated benchmark aimed at studying the ability of MT systems to choose the correct sense when translating an ambiguous source word occurring in a given context. Our benchmark requires a model to translate a sentence containing an ambiguous source word and evaluates the correctness of its translation.

We first illustrate the composition of the benchmark as well as its creation process consisting of both automatic and manual steps. Subsequently, we detail the evaluation procedure and the metrics used in the DIBIMT benchmark.

3.1 Composition of the Benchmark

Each instance in the benchmark is defined as a tuple $i = (s, w, \sigma, \mathcal{G}, \mathcal{B})$ composed of the following elements:

i. one source sentence s containing an ambiguous source word w which is associated with a given meaning σ. In our benchmark, due to its predominance in MT, we pivot on the English language and use it as the reference language for all our source sentences. The meaning σ corresponds to a synset in BABELNET[4] (Navigli and Ponzetto 2010; Navigli et al. 2021), the largest multilingual encyclopedic dictionary

4 https://babelnet.org.

Figure 1
Example of a benchmark instance. The focus word is **shot**, in its meaning of a "small drink of liquor." We expect correct translations to be, for example, in Italian *goccio* (small quantity), but not, for example, in German, *Injektion* (injection).

which follows and extends the WordNet synset model, where a given concept is represented through a set of lexicalizations which are used in different languages to express such concept.

ii. a set of good translations[5] $\mathcal{G}$ for the ambiguous word w in the context of s.

iii. a set of bad translations[6] $\mathcal{B}$ for the ambiguous word w in the context of s.

We note that the ambiguous source word w, and both GOOD and BAD translations, can be multi-word expressions and compounds. Furthermore, translations can have a different part of speech than the corresponding source word. As target languages, that is, the languages of the GOOD and BAD translations for each source word, DIBIMT covers: Bulgarian, Chinese, Dutch, German, Italian, Russian, Slovene, and Spanish. Figure 1 illustrates examples of GOOD and BAD translations for the instance (*he poured a shot of whiskey*, *shot*, *bn:00057755n*[7], {trago$_{\text{ES}}$, chupito$_{\text{ES}}$, goccio$_{\text{IT}}$, ..., glaasje$_{\text{NL}}$}, {Schlag$_{\text{DE}}$, sparo$_{\text{IT}}$, prik$_{\text{NL}}$, ..., выстрел$_{\text{RU}}$}).

3.2 Creation Process

3.2.1 Automatic Extraction of Instance Candidates. A candidate instance $i = (s, w, \sigma, \mathcal{G}, \mathcal{B})$ of our benchmark is extracted from two lexical-semantic resources, namely, WordNet

5 We refer to these translations with the tag GOOD.
6 We refer to these translations with the tag BAD.
7 BABELNET synset id corresponding to the meaning of *a small drink of liquor*, https://babelnet.org/synset?id=bn:00057755n&lang=EN.

(Miller 1995) and the English Wiktionary.[8] As far as WordNet is concerned, we consider each sentence s from the Princeton WordNet Gloss Corpus (Langone, Haskell, and Miller 2004) that is a usage example of a word w manually tagged with a specific WordNet synset. For Wiktionary, we extract every usage example s of a word w, excluding archaic usages and slang. We use the linkage between WordNet synsets (or Wiktionary definitions) and BABELNET synsets to provide a synset annotation σ for our instance i.

We are now left with the task of populating the set of GOOD and BAD translations $\mathcal{G}$ and $\mathcal{B}$ for each instance i in all eight target languages. Given the BABELNET synset σ in instance i, we define $\Lambda_L(\sigma)$ as the set of lexicalizations of σ in language L contained within synset σ. For instance, let us consider the synset $\tilde{\sigma}$ of *duck* defined as *broad-billed swimming bird*. $\tilde{\sigma}$ contains lexicalizations in different languages such as: *Ente* in German, *anatra* in Italian, *утка* in Russian, and *pato* in Spanish. Hence, $\Lambda_{\text{EN}}(\tilde{\sigma}) = \{duck\}$, while $\Lambda_{\text{IT}}(\tilde{\sigma}) = \{anatra, anitra\}$. We then define $\mathcal{G}_L$ and $\mathcal{B}_L$ as the set of GOOD and BAD translations in the target language L. We pre-populate these two sets automatically as follows. We assign $\mathcal{G}_L = \Lambda_L(\sigma)$, that is, the set of senses in language L derived from the BABELNET synset σ. Instead, we consider as BAD translations all the lexicalizations in language L of any other synset containing w, except those in $\mathcal{G}_{\mathcal{L}}$.

In order to create challenging instances, we adopt the following sentence filtering procedure. First, we retain only instances including an ambiguous word. Second, we retain at most one sentence per sense per source (WordNet or Wiktionary), so as to allow for semantic heterogeneity in the resulting dataset. Third, given the set of languages considered S_l, we retain instances $(s, w, \sigma, \mathcal{G}, \mathcal{B})$ which satisfy the following constraint:

$$\forall L \in S_l, \forall g \in \mathcal{G}_L \nexists\, \sigma' s.t.\ \sigma \neq \sigma', w \in \Lambda_{\text{EN}}(\sigma'), g \in \Lambda_L(\sigma') \tag{1}$$

For example, consider an instance with $w = bank$ and σ defined as *a flight maneuver*. $\Lambda_{IT}(\sigma)$ includes the lexicalization *avvitamento*, which is ambiguous in Italian, as it can refer either to the *laying of a screw* or a *downward spiral* (e.g., *the downward spiral of costs*), among other meanings. Therefore, to satisfy our third requirement, we ensure that no other possible senses of *avvitamento* (i.e., found in a different synset σ′) can be translated with *bank* into English according to BABELNET.

3.2.2 Manual Validation of the Extracted Instances. As a result of the aforementioned automatic process, we extract 948 candidate sentences. We now proceed to manually validate the entire dataset resulting from the previous procedure. To this end, we employ six annotators with proven experience in lexical semantics and MT.[9]

Annotators are asked to validate both $\mathcal{G}_L$ and $\mathcal{B}_L$, which contain the set of GOOD and BAD translations for the ambiguous source word w, respectively. Concerning GOOD translations, annotators are asked not only to validate the candidates proposed automatically, but also to include additional translations. Instead, regarding BAD translations, annotators are only required to establish the incorrectness of the automatically extracted translations. To determine the correctness of a candidate translation for w, annotators are required to evaluate its suitability to the context provided, regardless of whether this translation pertains to a part of speech other than that of w. Furthermore, annotators are

8 We use WordNet 3.0 and the Wiktionary dump of September 2021.

9 Annotators are either native speakers, hold C2-level certifications, or work as professional translators in the given language combinations. All annotators were either hired to carry out this task or are co-authors of the present article.

Table 2
Annotation statistics: %OG corresponds to the average percentage of Good lemmas which are Original (i.e., added by our annotators); %RG is the average percentage of Good lemmas that were Removed (i.e., lemmas derived from BABELNET that our annotators deemed incorrect in the context of the given instance).

	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
%OG	59.7	45.3	58.9	70.7	53.0	91.7	54.6	97.2	66.4
%RG	44.4	21.8	47.1	49.8	62.9	91.3	64.5	95.3	59.6

instructed to analyze the automatically collected sentences as well as the ambiguous source words and verify compliance with the requirements, among which we highlight that: (i) each sentence must provide a semantic context such that the meaning of the ambiguous source word w can be determined unequivocally; (ii) w cannot be part of a complex lexical unit such as an idiomatic expression; and (iii) the sentence is complete (e.g., not just a phrase). We report the complete annotation guidelines adopted in Appendix A.

As a result of the aforementioned manual process, approximately 30% of the 948 instances are discarded due to non-compliance with the guidelines, resulting in 667 instances. As shown in Table 2, on average, 66.4% of the correct translations are added manually by the annotators, whereas only 33.6% are derived automatically from BABELNET. Moreover, on average, almost 60% of the synset lexicalizations in a given language cannot be considered as GOOD translations for the corresponding ambiguous word in each instance. On the one hand, these statistics confirm that synonyms cannot be used interchangeably in translation. On the other hand, they are evidence of the noise contained in automatic lexical semantic resources such as BABELNET. This suggests that high quality in benchmarks such as DIBIMT cannot be achieved by merely relying on automatic resources and without manual intervention.

The summary statistics of the annotated dataset are reported in Table 3. DIBIMT covers 525 different synsets and 358 English lemmas, constituting a total of 667 manually curated instances. The benchmark is available in eight language combinations, that is, from English to each of the following languages: Bulgarian, Chinese, Dutch, German, Italian, Russian, Slovene, and Spanish. Furthermore, we compute the distribution of BABELNET domains over the synsets included in our instance set. As shown in Figure 2, DIBIMT is a heterogeneous benchmark, covering a wide range of domains.

Table 3
General statistics of our annotated dataset. POS-specific lemmas do not sum to "All" as they can overlap across POS tags (e.g., run). Poly. Deg. and Sense Freq. represent the average polysemy degree (i.e., number of meanings a word can have) and sense frequency index (i.e., how frequent a meaning is for a given word) of each DIBIMT instance.

	All	Nouns	Verbs
# items	667	350	317
# lemmas	358	219	175
# synsets	525	284	241
Poly. Deg.	10.56	9.87	11.33
Sense Freq.	5.38	4.08	6.03

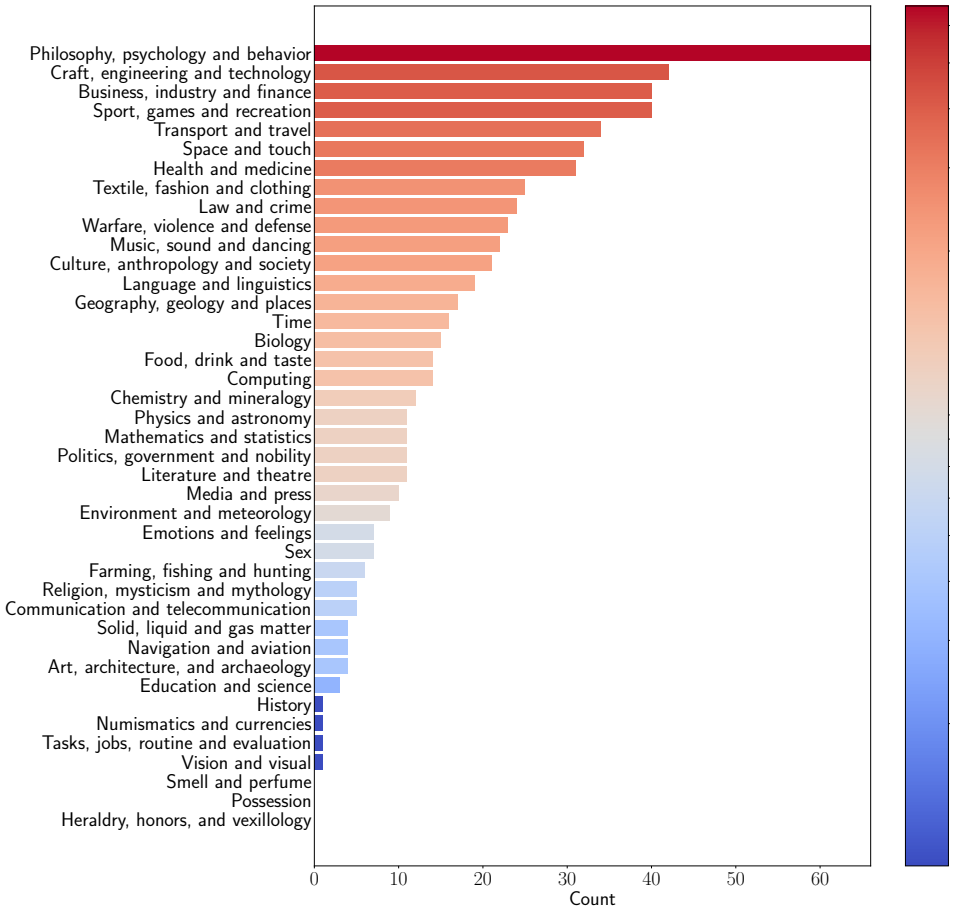


Figure 2
Coverage of domains in DIBIMT calculated as the absolute frequency of BABELNET domains (Camacho-Collados and Navigli 2017, and further extended in `https://www.babelnet.org/how-to-use`) associated with the synsets of the focus words occurring in all source sentences.

3.3 Evaluation Procedure

We now describe the automatic procedure that we adopt to determine the correctness of a translation provided by a candidate MT model according to DIBIMT. Given a DIBIMT instance i and a translation τ of sentence s produced by a candidate model into language L, we use Stanza (Qi et al. 2020) to perform tokenization, part-of-speech tagging, and lemmatization of τ. The analysis procedure computes the lexical overlap between the set of lemmas in τ and the two sets of manually validated translations, that is, $\mathcal{G}_L$ and $\mathcal{B}_L$. When dealing with multi-word expressions in `GOOD` or `BAD` translations, annotators are allowed to use wildcards so as to enable matching of non-contiguous spans (e.g., *auf * Wünsche eingehen* which means *to attend to somebody's wishes*, where * represents any sequence of words). Additionally, since Stanza features multi-word expansion tokenization for some of the languages considered, whenever possible, we perform matching on both the list of words and tokens in the translated sentence.

Finally, we label a given instance as: (i) MISS if no match is found; (ii) UNK in the case that both GOOD and BAD translation sets match; and (iii) GOOD or BAD depending on which set matches a given lemma contained in the translation produced by the MT model.

3.3.1 Improving the Benchmark Matching Capability. By manually inspecting misclassified instances, we notice that one of the main causes is to be found in textual pre-processing steps such as tokenization, lemmatization, and part-of-speech tagging. In order to minimize pre-processing errors and reduce MISS tags, we apply multiple Stanza models whenever available for the same language,[10] owing to the fact that each model is trained on a specific treebank from Universal Dependencies (Nivre et al. 2016), and each treebank adopts its own domain and annotation guidelines. Specifically, we use the following packages: GSD and HDT for German; AnCora and GSD for Spanish; Combined,[11] ISDT, ParTUT, VIT, TWITTIRO, and PoSTWITA for Italian; SyntagRus, GSD, and Taiga for Russian; GSDSimp[12] for Chinese; BTB for Bulgarian; Alpino and LassySmall for Dutch; and SSJ, SST, and, additionally, CLASSLA for Slovene (Ljubešić and Dobrovoljc 2019).

3.3.2 Language-specific Rules. In addition to the aforementioned improvements, we enrich our analysis procedure with auxiliary heuristics dealing with specific linguistic phenomena, such as compounds and reflexive verbs, and aimed at further reducing the MISS instances.

Reflexive Verbs. Preliminary analyses show that the standard evaluation procedure fails to recognize GOOD or BAD translations represented by reflexive verbs in languages such as German, Italian, and Spanish. For the latter two languages, we address these cases as follows. First, we identify reflexive verbs included as GOOD and BAD translations by inspecting the ending, that is, *rsi* and *rse* in Italian and Spanish, respectively. Subsequently, we associate such reflexive verbs with some reflexive pronouns in the corresponding language, that is, *mi, ti, si, ci, vi* in Italian and *me, te, se, nos, os* in Spanish. Finally, we determine whether the lemmatized sentence produced by a given model contains the reflexive form included as GOOD or BAD or the corresponding non-reflexive base form of the verb and one of the associated reflexive pronouns.

Instead, in German we check whether a given GOOD or BAD translation is composed of at least two elements, the first of which is the reflexive pronoun *sich*. If this is the case, we look for a match between the base form of the verb that comes after *sich* and any of the reflexive pronouns *mich, dich, sich, uns, euch* in the lemmatized candidate translation. We leave it to future work to address multiple cases and scenarios as well as to extend and improve these rules to all languages covered.

German-specific Rules. The correct detection of a GOOD or BAD translation into German requires additional rules regarding specific linguistic phenomena such as word order, multi-word expressions, compounds (Komposita), and separable verbs. For instance, given the source sentence *He thought Nehru jackets* ***went out*** *of fashion at the end of the seventies* translated into German with *Er dachte, Nehru-Jacken* ***kamen*** *Ende der siebziger*

10 We follow an iterative approach, where the entire analysis procedure is executed with each package but only considering the remaining MISS instances.

11 A package trained on a concatenation of ISDT, VIT, PoSTWITA, and TWITTIRO. We perform the iterative analysis with the single-dataset models as well in order to increase redundancy.

12 GSD package converted to simplified characters.

*Jahre **aus der Mode***, the correct lemmatization of *kamen [...] aus der Mode* is *aus der Mode kommen*. In order to ensure that the GOOD translation *aus der Mode kommen* is detected in the aforementioned translation, we implement a multi-word inversion mechanism capable of dealing with these cases.

Another challenge when translating into German concerns separable verbs, such as *zurücklegen*. For example, given the source text *The caravan **covered** almost 100 miles each day* translated with *Die Karawane **legte** jeden Tag fast 100 Meilen **zurück***, the correct lemmatization of *legte [...] zurück* is *zurücklegen*. To address these specific cases, we rely on the dependency tree, which classifies the nature of the relationship between prefixes of separable verbs and the main part of the verb using the Universal Dependencies[13] dependency relation *compound:prt*. Specifically, we combine the prefix with the lemmatized core part of the verb, obtaining forms such as *zurücklegen*.

3.3.3 Handling UNKs. We observe that, in some cases, instances are classified both as GOOD and BAD; we define these cases as UNK instances. Interestingly, our manual inspection shows that UNK instances are often caused by the match between a single word w appearing in $\mathcal{B}$ and a compound word, or multi-word, containing w, found within $\mathcal{G}$. We address these cases by attributing higher priority to the identification of the longest matches, thereby ignoring potential shorter matches contained in longer matches already classified as either GOOD or BAD. Importantly, we note that UNK instances represent, on average, 1.15% of all DIBIMT items, thus having a limited impact on the overall evaluation of semantic biases. We report the breakdown of UNK instances in Appendix D.1.

3.4 Metrics of the DIBIMT Benchmark

In this section, we detail the metrics used in the DIBIMT benchmark to evaluate MT models. In addition to accuracy, DIBIMT analyzes the semantic biases of a translation model via four metrics, which we illustrate in what follows. To formally define these metrics, we adopt the following notation:

- λ_P represents a (lemma, part of speech) pair, where P is the part of speech;
- $\Omega_L(\lambda_P) = \{\sigma_1, \ldots, \sigma_n\}$ indicates the set of synsets pertaining to the part of speech P which contain λ as a lexicalization in language L;
- $\mu_{\lambda_P}(\sigma)$ refers to the index of synset σ in $\Omega_{\text{EN}}(\lambda_P)$ ordered according to WordNet's sense frequency, as computed from SemCor. That is, index k means that synset σ is the k-th most frequent meaning for λ_P.

All the metrics defined in this section are computed on the set of translations produced by a given model $\mathcal{M}$ in a target language L, excluding MISS and UNK occurrences.

Accuracy. The standard accuracy metric provides a general view of a model's performance in a given language. It is computed as: $\frac{\#\text{GOOD}}{\#\text{GOOD}+\#\text{BAD}}$.

Sense Frequency Influence (SFI). This metric assesses the impact of frequency on MT models' ability to handle ambiguous words. First, we group instances based on their

13 https://universaldependencies.org/.

ambiguous word meaning's frequency index, that is, $\mu_{\lambda_P}(\sigma)$. We define $\texttt{\#GOOD}_\mu$ as the number of GOOD instances whose ambiguous word's meaning has a frequency index of μ; similarly, $\texttt{\#BAD}_\mu$ is defined for BAD instances. Moreover, we introduce the set of all frequency indices a synset can assume in the DIBIMT dataset as $M = \{\mu_{\lambda_P}(\sigma) | \lambda_P \in \text{LP}, \sigma \text{ is the meaning of } \lambda_P \text{ in context}\}$, where LP represents the set of all (lemma, part of speech) pairs corresponding to the target words in DIBIMT.

We propose the Sense Frequency Influence (SFI) as a measure of accuracy, where the weight of each instance is adjusted based on its sense frequency index:

$$\text{SFI} = \frac{\sum_{\mu \in M} \mu \cdot \texttt{\#GOOD}_\mu}{\sum_{\mu \in M} \mu \cdot \left(\texttt{\#GOOD}_\mu + \texttt{\#BAD}_\mu\right)} \tag{2}$$

As a result, instances focused on lower-frequency senses are given more prominence in the SFI metric. We discuss our choice of using frequency index-based weighting instead of frequency-based weighting in Appendix B.

Polysemy Degree Importance (PDI). This metric investigates the impact of the polysemy degree of words on the disambiguation capabilities of a model. To this end, we define $\delta_L(\lambda_P) = |\Omega_L(\lambda_P)|$ as the polysemy degree, that is, the number of senses of λ_P in language L. This metric mirrors SFI in the way it is computed, but groups instances by their ambiguous word's polysemy degree $\delta_{\text{EN}}(\lambda_P)$ instead of μ.

Most and More Frequent Senses. Finally, we measure the frequency with which models predict senses that are more common than the correct one. Given an instance classified as BAD, we denote $\hat{\sigma}$ as the synset associated with the incorrectly translated lemma.[14] Subsequently, we compute the frequency index of σ and $\hat{\sigma}$ with respect to λ_P. If $\mu_{\lambda_P}(\hat{\sigma}) < \mu_{\lambda_P}(\sigma)$, then the system selected a sense which is more frequent than the correct one and which we call More Frequent Sense (MFS+). Additionally, if $\mu_{\lambda_P}(\hat{\sigma}) = 1$, then the model disambiguated the source word w to the Most Frequent Sense (MFS) of the associated lemma λ_P.

4. Performance on the DIBIMT Benchmark

We now put our DIBIMT framework into practice, analyzing and discussing the performance obtained by state-of-the-art MT systems and LLMs. First, we describe the systems considered. Then, we detail critical issues that surfaced as a result of preliminary experiments as well as our strategies to address them. Lastly, we illustrate our results.

4.1 Evaluated Systems

We evaluate a wide range of systems, including commercial and non-commercial MT models and LLMs. Specifically, we test the following MT systems:

- **Google Translate,**[15] one of the most popular commercial MT systems.

14 If there are multiple synsets associated, we consider the most frequent one, that is, the one with the lowest frequency index according to $\mu_{\lambda_P}(\cdot)$.

15 https://translate.google.com/.

- **DeepL Translator**,[16] a state-of-the-art commercial MT system.
- **MBart50** (Tang et al. 2021), multilingual BART fine-tuned on the translation task for 50 languages (around 610M parameters). We refer to **MBart50** as the English-to-many model, and to **MBart50$_{\text{MTM}}$** as the many-to-many model.
- **M2M100** (Fan et al. 2021), a multilingual model able to translate from/to 100 languages. We evaluate both versions of the model, the one with 418M parameters (**M2M100**) and the one featuring 1.2B parameters (**M2M100$_{\text{LG}}$**).
- **OPUS** (Tiedemann and Thottingal 2020), one of the smallest state-of-the-art NMT models available to date; a base Transformer (each model features approximately 74M parameters) trained on a single language pair on large amounts of data. Moreover, we evaluate its multilingual counterpart, jointly trained on multiple target languages, which we dub **OPUS$_{\text{MUL}}$**.[17]
- **NLLB-200** (NLLB Team et al. 2022), a state-of-the-art MT system trained on approximately 200 different languages. NLLB-200 leverages multiple data augmentation techniques, such as back-translation via both neural and statistical MT models, and high-quality bitext mining. Specifically, we consider the following four different versions of **NLLB-200**:
  - **NLLB-200$_{\text{SM}}$**: smallest version, distilled, counting 600M parameters;
  - **NLLB-200$_{\text{MD}}$**: medium version, counting 1.3B parameters;
  - **NLLB-200$_{\text{MDD}}$**: medium version, distilled, counting 1.3B parameters;
  - **NLLB-200$_{\text{LG}}$**: large version, the baseline model presented in NLLB Team et al. (2022), counting 3.3B parameters.

Furthermore, we evaluate the following LLMs:

- **Gemma 1** (Gemma Team et al. 2024a), a family of open-source language models based on Google's Gemini (Gemini Team et al. 2024). Gemma is based on the Transformer decoder architecture (Vaswani et al. 2017), relying on crucial improvements such as the MultiQuery Attention (Shazeer 2019), rotary positional embeddings (Su et al. 2024), and the GeGLU activation function (Shazeer 2020). We use:
  - **Gemma-2B** (`google/gemma-2b-it`), which features 2B parameters and is trained on 3T tokens;

16 `https://deepl.com/`.

17 We use `Helsinki-NLP/opus-mt-en-mul` from HuggingFace. The bilingual models' signature instead is `Helsinki-NLP/opus-mt-en-xx`, with `xx` varying across the languages supported by DIBIMT, except for Slovene, which is not available.

  - **Gemma-7B** (`google/gemma-7b-it`), which is a 7B model trained on 6T tokens.

- **Gemma 2**, that includes new additional Gemma models which range in scale from 2B to 27B parameters. We use **Gemma2-9B** (`google/gemma-2-9b-it`), with 9B parameters and trained on 8T tokens (Gemma Team et al. 2024b). Compared to Gemma 1 models, Gemma2-9B additionally utilizes the interleaving local-global attentions (Beltagy, Peters, and Cohan 2020) and the Grouped-Query Attention (Ainslie et al. 2023) and is trained using knowledge distillation by minimizing the negative log-likelihood between the probabilities of a larger model, the teacher, and a smaller one, the student.
- **LLaMA 2**, a family of models developed by Meta (Touvron et al. 2023). Trained on publicly available datasets only, LLaMA 2 models range from 7 to 65 billion parameters. LLaMA 2 models rely on the Transformer architecture, with improvements such as the SwiGLU activation function (Shazeer 2020) and rotary embeddings. We use **LLaMA2-7B** (`meta-llama/Llama-2-7b-chat-hf`), a 7 billion parameter conversational model optimized for generating human-like responses in a dialogue setting.
- **LLaMA 3**, an improved version of the LlaMA 2 model family. LLaMA 3 models are available in two sizes: 8 billion and 70 billion parameters. One notable enhancement in LLaMA 3 is the implementation of a new tokenizer, which significantly increases the vocabulary size to 128,256 tokens, compared with the 32,000 tokens used in LLaMA 2. We evaluate `meta-llama/Meta-Llama-3-8B-Instruct`, which we dub **LLaMA3-8B**.
- **Mistral-7B**, a pretrained LLM featuring 7 billion parameters (Jiang et al. 2023). Its vocabulary size is 32,000. We evaluate `mistralai/Mistral-7B-Instruct-v0.2`.
- **Tower-7B**, a 7 billion parameter model fine-tuned for translation-related tasks, such as MT, automatic post-editing, and Named Entity Recognition (Alves et al. 2024). We experiment with `Unbabel/TowerInstruct-7B-v0.1`. Built on top of LLaMA 2, this model has been fine-tuned to translate between English and the following languages: Chinese, Dutch, French, German, Italian, Korean, Portuguese, Russian, and Spanish.
- **Phi-3**, a family of small language models by Microsoft (Abdin et al. 2024). Phi-3 language models are available in both short- and long-context lengths. We evaluate `microsoft/Phi-3-mini-128k-instruct`, a lightweight language model with 3.8B parameters, supporting a context length of 128K tokens. `microsoft/Phi-3-mini-128k-instruct` is trained on 3.3T tokens and its vocabulary size is 32,064.
- **GPT-3.5$_{\text{TURBO}}$**, a large language model optimized for chat applications. The translations were produced by the default model at the time of writing, that is, `gpt-3.5-turbo-0613`.
- **GPT-4**, a large-scale multimodal language model that achieves better performance than its competitors (OpenAI et al. 2024). The translations

from GPT-4 were produced by the default model at the time of writing, that is, `gpt-4-0613`.[18]

We report in Appendix C the details about the generation parameters we use for each open-source MT system, together with the prompt provided to the LLMs.

4.2 Evaluation Metrics

To test the aforementioned systems, we adopt the evaluation procedure illustrated in Section 3.3, using the metrics we defined in Section 3.4. Thus, each system is assigned specific scores to account for different aspects of the evaluation, that is, its overall accuracy, its ability to deal with infrequent senses and highly ambiguous words, represented by SFI and PDI metrics, respectively, and the frequency with which it predicts senses that are more common than the correct one, represented by the MFS and MFS+ metrics.

4.3 Benchmark Refinement

In order to reduce the number of `MISS` instances we perform a manual refinement, which we illustrate in this section. Specifically, annotators are required to inspect the `MISS` instances resulting from the evaluation of some of the considered systems and determine whether these should be classified as `GOOD` or `BAD`. This step is prompted by the consideration that the set of `GOOD` and `BAD` translations is not closed and can always be augmented with new suitable items. We highlight that, due to time and budget constraints, the aforementioned refinement step does not involve the NLLB-200 models, OPUS_{MUL}, and the considered LLMs.

In Table 4 we report the average percentage of `MISS` instances on the DIBIMT benchmark before and after the `MISS` refinement, and the related accuracy scores. As we can see, the refinement process substantially reduces the percentage of `MISS` instances, with an average reduction of 19.75% across all tested systems. Furthermore, we notice that the refinement produces an average accuracy increase of 19.61%, suggesting that a `MISS` instance might frequently be associated with a correct translation, rather than an incorrect one. From these results, we can also notice that manually inspecting sentences produced by some models does not bias the benchmark toward them. Indeed, the average drop in `MISS` % in refined and non-refined systems is 22.82% and 18.31%, respectively, and the average accuracy increase is 18.49% and 20.14%, respectively, with very similar deltas between the two groups.

However, we wish to point out two notable exceptions. The first concerns OPUS_{MUL}, for which we report a substantially smaller `MISS` % drop and accuracy increase, compared with other models. We attribute this to its particularly low accuracy, as will be discussed in the next section. Indeed, systems struggling to translate ambiguous focus words often generate contextually inappropriate tokens or severe hallucinations, which arguably should not be included in the benchmark as they are not disambiguation errors. We report examples of these phenomena in the qualitative analysis of Section 5. The second exception is represented by the open-access LLMs. As we can see from Table 4, these systems display a much higher percentage of `MISS` instances than the others. We attribute this to the fact that, to the best of our knowledge, they perform MT in

18 Translations were generated in November 2023.

Table 4
Impact and generalization of manual MISS refinement, comparing refined and non-refined models, in terms of average MISS % and accuracy score across languages. Full tables with per-language breakdowns are reported in Appendix G. The final mean accuracy after refinement differs from Table 5 as it is computed over model averages.

		MISS %			Accuracy		
	Model	Before	After	Δ (−)	Before	After	Δ (+)
Refined	**Google**	49.41	15.61	33.80	33.88	56.85	22.97
	DeepL	53.36	15.31	38.04	40.08	65.66	25.59
	MBart50	57.47	41.87	15.60	17.88	35.49	17.61
	MBart50$_{MTM}$	52.27	32.31	19.96	18.11	35.45	17.35
	M2M100	51.24	33.12	18.12	11.13	25.83	14.70
	M2M100$_{LG}$	52.43	36.02	16.42	15.69	31.78	16.09
	OPUS$_{BIL}$	44.22	26.41	17.81	20.86	35.98	15.12
	Mean	51.49	28.67	22.82	22.52	41.01	18.49
Non-refined	**OPUS$_{MUL}$**	61.28	52.24	9.04	9.08	20.06	10.98
	NLLB-200$_{SM}$	55.01	37.23	17.78	21.67	40.55	18.88
	NLLB-200$_{MD}$	57.99	36.86	21.13	29.64	51.33	21.68
	NLLB-200$_{MDD}$	56.87	34.20	22.67	33.02	53.92	20.90
	NLLB-200$_{LG}$	57.68	33.43	24.26	36.40	57.82	21.42
	Llama2-7B	65.97	42.20	23.78	25.68	55.45	29.77
	Llama3-8B	60.53	53.58	6.95	33.70	46.05	12.35
	Mistral-7B	62.04	45.18	16.87	27.13	48.31	21.18
	Gemma-2B	71.82	63.06	8.75	23.40	40.53	17.13
	Gemma-7B	63.91	50.60	13.32	26.94	46.79	19.85
	Gemma2-9B	61.46	36.64	24.83	45.38	69.80	24.42
	Phi3-mini	67.18	57.42	9.76	24.36	42.57	18.22
	Tower-7B	61.18	42.82	18.36	40.60	60.55	19.95
	GPT-3.5$_{TURBO}$	55.47	27.87	27.60	44.43	67.30	22.87
	GPT-4	54.77	25.16	29.61	48.66	71.09	22.43
	Mean	60.88	42.56	18.31	31.34	51.47	20.14
	Mean	57.89	38.14	19.75	28.53	48.14	19.61

a zero-shot setting, without prior fine-tuning for the task.[19] Indeed, it has already been shown that the zero-shot translation performance of most LLMs can be substantially improved by either using few-shot learning (Zhang, Haddow, and Birch 2023; Bawden and Yvon 2023), or by fine-tuning them to MT (Alves et al. 2024; Xu et al. 2024). Also, all the tested LLMs were pre-trained using mostly English data, with very limited exposure to some of the target languages in the DIBIMT benchmark. As a consequence of the foregoing factors, open-access LLMs frequently hallucinate and produce omissions. We delve deeper into this matter in our linguistic analysis, in Section 5.

Finally, we report a total average of 38.14% MISS instances. We wish to highlight that this number is inflated by the high percentage of MISS instances of the open-access LLMs. Indeed, considering only the other systems, the average MISS % is reduced to

19 Tower-7B is an exception. Nonetheless, it was not fine-tuned to translate toward Bulgarian and Slovene, that is, the language directions with the highest MISS %, which is almost double compared with that of the other language directions (Table 23).

31.97%. Additionally, we argue that some translations are ultimately non-classifiable, for example, a complete omission of the target word could still be correct if the translation conveys the meaning of the source sentence, as we show in the linguistic analysis of Section 5. Finally, we believe that manually accounting for any possible translation produced by all existing models is unfeasible, due to the high variability intrinsic to the translation task.

4.4 Results

We report the accuracy of the aforementioned systems on the DIBIMT benchmark in Table 5. We observe average values ranging from 20.06% to 71.09%, with an absolute low of 9.76% (OPUS_{MUL} in Chinese), and an absolute high of 83.37% (Gemma2-9B in Russian).

The best-performing system is GPT-4, with an average accuracy of 71.09%, followed by Gemma2-9B and $\text{GPT-3.5}_{\text{TURBO}}$. We note that the largest (and more recent) LLMs achieve competitive results despite their not being fine-tuned directly for MT. This confirms the results of Iyer, Chen, and Birch (2023), where the authors found

Table 5
Systems' accuracy on the DIBIMT benchmark. For each translation direction, the best score is bold and underlined, the second best is bold, and the third best is underlined.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	62.42	57.49	57.94	68.34	56.61	50.84	44.79	56.39	56.85
DeepL	**81.92**	64.87	70.66	73.73	55.52	62.69	52.44	63.47	65.66
MBart50	34.29	31.57	38.38	40.86	39.35	–	23.53	40.48	35.49
$\text{MBart50}_{\text{MTM}}$	34.20	34.81	38.29	40.05	39.08	–	22.77	38.97	35.45
M2M100	26.25	27.62	26.62	31.84	20.39	22.64	18.90	32.41	25.83
$\text{M2M100}_{\text{LG}}$	31.84	32.78	34.39	38.58	27.36	29.29	23.97	36.02	31.78
OPUS_{BIL}	34.88	37.39	37.15	41.29	34.45	35.71	31.01	–	35.98
OPUS_{MUL}	21.08	25.58	21.72	24.05	9.76	14.87	18.48	24.91	20.06
$\text{NLLB-200}_{\text{SM}}$	38.03	42.21	43.32	50.86	38.87	37.08	29.35	44.71	40.55
$\text{NLLB-200}_{\text{MD}}$	54.02	51.88	55.38	58.97	45.00	47.80	41.18	56.37	51.33
$\text{NLLB-200}_{\text{MDD}}$	58.37	51.87	59.21	59.26	48.93	50.83	44.58	58.29	53.92
$\text{NLLB-200}_{\text{LG}}$	63.73	55.74	63.85	66.52	47.40	54.29	49.05	61.97	57.82
Llama2-7B	48.82	49.43	51.63	53.28	45.52	33.98	37.93	47.83	46.05
Llama3-8B	64.16	56.35	57.68	67.77	55.65	40.31	46.60	55.09	55.45
Mistral-7B	58.13	49.77	54.91	57.45	44.57	36.14	38.78	46.72	48.31
Gemma-2B	45.51	46.65	39.93	51.16	43.43	27.50	27.41	42.62	40.53
Gemma-7B	50.69	51.03	52.37	59.26	51.19	33.03	34.49	42.29	46.79
Gemma2-9B	74.72	65.70	**72.76**	**83.37**	**68.76**	**64.94**	60.90	67.23	**69.80**
Phi-3-mini	61.01	52.25	50.82	48.79	39.87	23.64	29.39	34.83	42.57
Tower-7B	70.31	64.83	66.73	72.30	58.02	43.71	57.04	51.45	60.55
$\text{GPT-3.5}_{\text{TURBO}}$	73.75	**66.48**	69.32	75.31	66.81	57.23	**61.41**	**68.11**	67.30
GPT-4	**77.22**	**70.04**	**71.71**	**80.20**	**71.15**	**64.16**	**62.69**	**71.55**	**71.09**
Mean	52.97	49.38	51.58	56.51	45.80	41.53	38.94	49.61	48.29

that instruction-tuned LLMs achieve state-of-the-art performance when dealing with disambiguation biases, with competitive scores even in unseen languages.[20] Interestingly, Gemma2-9B achieves an accuracy close to that of GPT-4, and better than one of the best commercial translation systems, that is, DeepL, despite having only 9B parameters. Also, Tower-7B, an LLM based on LLaMA2-7B additionally fine-tuned for MT-related tasks, achieves performance that is competitive with the larger LLMs and commercial MT systems.[21] This suggests that a promising direction for MT research could be eliciting the translation performance of LLMs. Indeed, sense (in)frequency poses one of the hardest challenges for MT systems (Section 4.4.1). Therefore, finding ways to leverage the information LLMs learn from monolingual corpora, which is abundant, might be key to mitigating this issue. In addition, thanks to the vast amounts of monolingual data available, LLMs can be scaled to accommodate higher parameter counts, learning more complex relations between words and their contexts. Indeed, the highest performance was obtained by GPT-4 which, most likely, is the largest system among the ones considered here.[22]

Among the open-source MT systems, instead, NLLB-200 variants outperform the others, with the larger NLLB-200_{LG} being the only open-source system that obtains results that are competitive with the commercial systems. Indeed, although on average commercial systems perform considerably better than their non-commercial counterparts, we observe that NLLB-200_{LG} manages to outperform Google Translate in every language except Spanish, Russian, and Chinese. The systems that attain the lowest accuracy are M2M100 and OPUS_{MUL}, that is, the smallest ones, together with OPUS_{BIL}. As expected, we observe that, for a given architecture, increasing the parameter count corresponds to an increase in accuracy; for instance, NLLB-200_{LG} obtains a mean accuracy of 57.82%, while NLLB-200_{SM} only 40.55% with a decrease of 17.27%, while $\text{M2M100}_{\text{LG}}$ outperforms M2M100 by 5.95%. Surprisingly, this does not apply for OPUS_{BIL}, whose overall accuracy is higher than that of OPUS_{MUL}, M2M100, $\text{M2M100}_{\text{LG}}$, MBart50, and $\text{MBart50}_{\text{MTM}}$, despite the last four being considerably larger. We defer the explanation of this phenomenon to Section 6.3.

Concerning the availability of resources in specific language pairs, we note that the mean accuracy for Chinese is particularly low, despite the large number of parallel corpora available for this translation direction. Furthermore, although Fan et al. (2021) consider German to be high-resource, compared with Slovene, M2M100 and $\text{M2M100}_{\text{LG}}$ exhibit better performance in Slovene—with accuracy scores of 32.41% and 36.02% — than they do in German, where their scores are 26.25% and 31.84%, respectively. The same pattern is observed in the outputs of OPUS_{MUL}. Finally, we highlight that, while substantial efforts are being made to improve the availability of parallel corpora for low-resource languages, we observe that, despite its low number of `MISS` instances, a high-resource language such as Dutch exhibits the poorest performance in DiBiMT (see Table 5). In light of this counterintuitive finding, we encourage the MT community to assess and address the impact of lexical ambiguity on high-resource languages as well, as it might still pose significant challenges for these.

20 Iyer, Chen, and Birch (2023) dub as *unseen* the languages not intentionally included in the pretraining of LLMs.

21 We highlight that Tower-7B's fine-tuning data did not include Bulgarian and Slovene, which is why its performance is lower in translating toward these languages.

22 Even if we ignore the exact number of parameters of the GPT systems, we can assume that they are at least one or two orders of magnitude larger than the other tested systems.

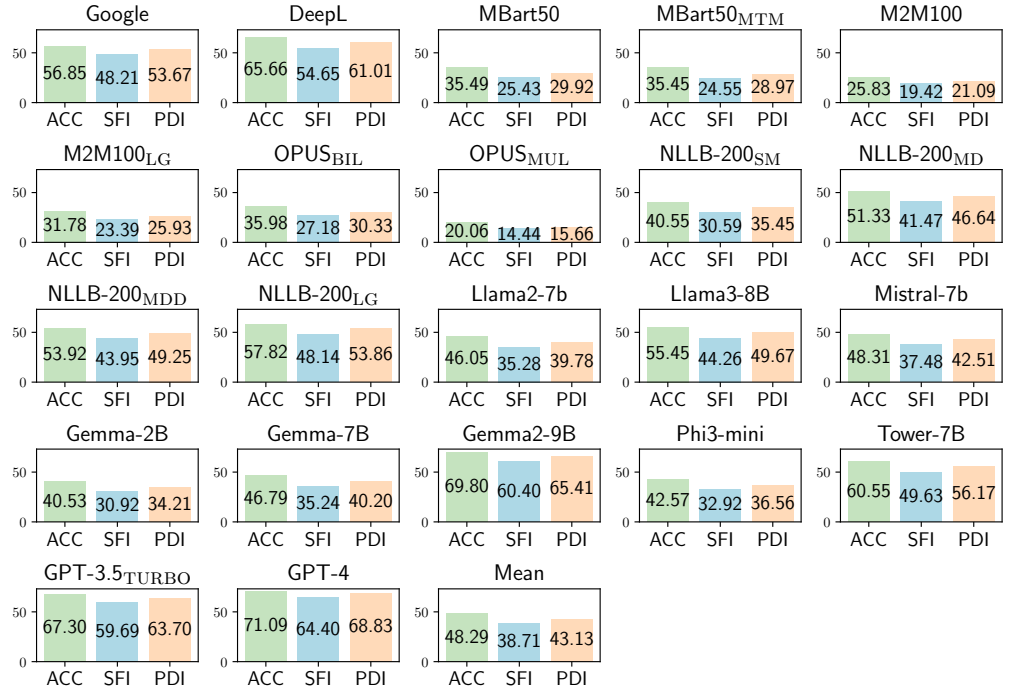


Figure 3
Mean accuracy, SFI, and PDI of each system on the DIBIMT benchmark. Full tables with per-language SFI and PDI scores are reported in Appendix G.

4.4.1 Identifying Pitfalls in Disambiguation. In Figure 3, we present the mean accuracy, SFI, and PDI scores for each system. As expected, both SFI and PDI exhibit lower values compared to accuracy, which is due to the systems' increased difficulty in accurately translating words with rare meanings and those that are highly ambiguous. Interestingly, SFI is consistently lower than PDI, suggesting that the infrequency of senses poses a particularly significant challenge for neural architectures.

To investigate this matter more thoroughly, we group the instances of DIBIMT based on their sense frequency index and polysemy degree, and present in Figure 4 the accuracy of each system on each identified group.[23] As anticipated, the accuracy significantly drops when the focus word exhibits a high polysemy degree, or its associated meaning is infrequent, and the decline in accuracy is more evident with increasing levels of frequency index, rather than polysemy degree. Interestingly, the most accurate systems demonstrate greater robustness to highly polysemous words. To illustrate this, we compare the average accuracy decrease when transitioning from words with a polysemy degree of less than 6 to those with 14 or more. The average drop in accuracy is 22.55 points; however, for top-performing systems (GPT-4, Gemma2-9B, GPT-3.5$_{\text{TURBO}}$, DeepL, Tower-7B, NLLB-200$_{\text{LG}}$, Google, LLaMA3-8B, NLLB-200$_{\text{MDD}}$, and NLLB-200$_{\text{MD}}$), this decrease is less marked, at 17.17 points, versus 27.03 points for the other systems.

23 We determined groups' ranges independently for each metric, aiming at populating each group with approximately 20% of the instances.

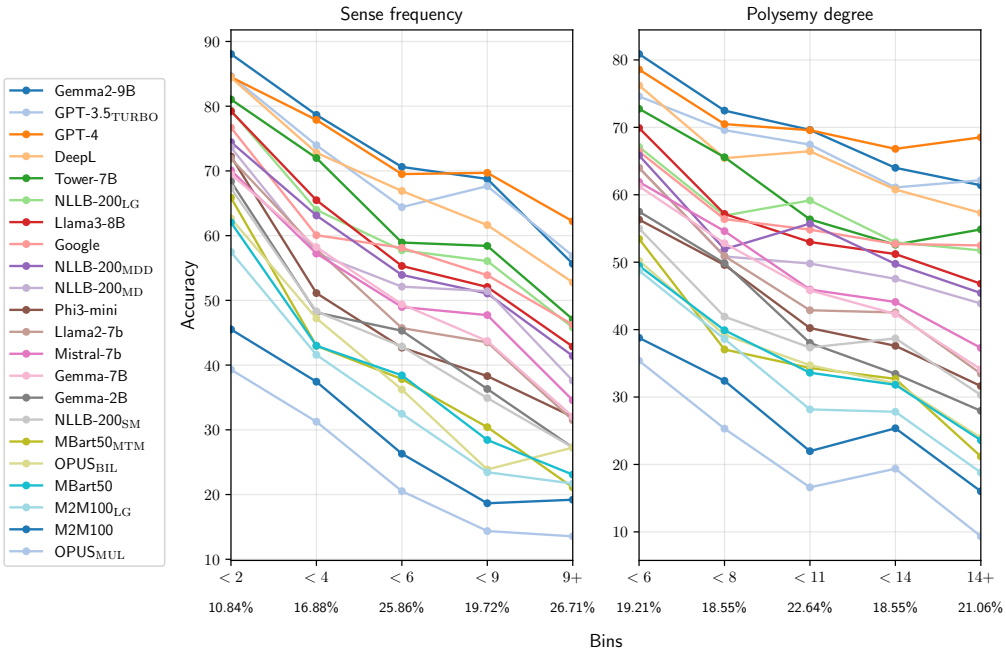


Figure 4
Accuracy of all systems when grouping the instances of the DiBiMT benchmark according to the sense frequency index and polysemy degree of their ambiguous focus words. Different groups have no instances in common, for example, the group with a sense frequency index < 6 is composed of those instances whose ambiguous focus words have a sense frequency index between 6 (excluded) and 4 (included). Below the x-axis, we report the percentage of instances that belong to each group.

We hypothesize that systems with limited contextualization capabilities struggle when selecting the correct translation from numerous possibilities. Instead, more accurate systems are trained on extensive data and have likely learned better decision boundaries for different senses of ambiguous words, making them more confident in picking translations representing the correct meaning, regardless of the number of possible translations.

Conversely, all systems seem to encounter similar difficulties when ambiguous words are used with particularly infrequent meanings, that are likely underrepresented in their training corpora. Indeed, the average accuracy drop between the most frequent sense (<2) and senses over the 9th frequency index is 34.51, and the gap between the average score of top- and bottom-performing systems is only 5.06, much lower than the gap we observe for PDI, at 9.86. In these cases, all systems tend to translate the focus word into one of its more common meanings. We delve deeper into this phenomenon in Appendix D, where Table 19 presents the scores of MFS and MFS+ metrics, which quantify how often a system translates the focus word into its most frequent sense (MFS), or a sense more frequent than the correct one (MFS+). On average, we report that the MFS is chosen more than half of the time a model translates incorrectly, whereas systems choose a more frequent sense more than 3 out of 4 times they make a mistake. We hypothesize that this tendency to translate into more frequent meanings is strictly

Table 6
Results by PoS tag. Numbers represent the mean value of each score introduced in the article. Column **All** summarizes the results reported in the other tables.

	All	Nouns	Verbs
Accuracy	48.29	47.80	48.78
% MISS	38.49	32.79	44.74
MFS	51.21	54.56	45.37
MFS+	76.44	75.95	77.88
SFI	38.71	39.15	38.24
PDI	43.13	42.87	43.37

related to the balance of senses in the training corpora. Indeed, since words' meanings appear in natural texts following a Zipfian distribution, neural models may tend to fall back upon this distribution whenever they are unsure about the meaning of an ambiguous word. In Section 4.4, we suggested exploring ways for leveraging monolingual corpora to counter the sense infrequency issue. In addition, we believe that another promising direction for MT research might be re-balancing the sense distribution found in monolingual and parallel corpora, to incorporate a larger number of occurrences of words used with infrequent meanings. However, we encourage caution and suggest using the DIBIMT benchmark to measure variations in the disambiguation accuracy of words with frequent and infrequent meanings. Indeed, modifying the sense distribution of MT systems' training data might have the undesirable outcome of trading off the disambiguation performance of frequent senses for infrequent ones.

4.4.2 Nouns vs Verbs. Notably, verbs are found to be harder to disambiguate than nouns (Barba, Procopio, and Navigli 2021), also considering the average polysemy degree of nouns and verbs in WordNet, namely, 1.24 and 2.17, respectively. We investigate whether this is true when evaluating systems against the DIBIMT benchmark. With this aim in view, Table 6 shows the average performance considered both on all parts of speech in DIBIMT, and on nouns and verbs separately. While the accuracy scores remain similar, we observe a significantly higher number of MISS instances for verbs. Interestingly, on average, the only notable difference seems to be that systems are more biased toward the most frequent sense when dealing with nouns. We attribute this to verbs having a higher average polysemy degree, and therefore a higher number of *more* frequent senses, making it more likely for systems to pick one or the other of these rather than homing in on the single most frequent one.

5. Manual Error Analysis

In this section, we address **RQ2**, that is, we investigate the presence of patterns when analyzing the errors produced by the systems under consideration when translating ambiguous focus words. As a result of our analysis, we identify the following four error patterns: (i) *disambiguation errors*, in which models fail to choose the correct sense, and, instead, select an incorrect sense, which is often more frequent than the correct one; (ii) *omissions*, when systems do not translate the focus word; (iii) *untranslated source words*, that is, the focus word is reproduced as-is in the translation; and (iv) *hallucinations*, when the focus word is translated with a word or expression that is semantically detached from the source, including when models generate a word or expression that does not

exist in the target language. In the following, we introduce and discuss these error patterns. Finally, we focus on errors encountered in general-purpose LLMs.

5.1 Disambiguation Errors

For the purposes of investigating the ability of MT systems to deal with lexical ambiguity, the most relevant error pattern to be studied is represented by disambiguation errors (also referred to as WSD errors), that is, translation errors in which systems do not select the correct sense for the ambiguous focus word.

In this error scenario, the incorrect sense chosen by a given system corresponds either to the most frequent meaning denoted by the focus word or simply a meaning that is more frequent than the correct one. For instance, the source text *The pitcher **delivered** the ball* is incorrectly translated into Dutch by Google with *De werper **leverde** de bal **af***, in which the verb *afleveren* does not convey the correct meaning of the focus word, that is, *throw or hurl from the mound to the batter*, as in baseball, which, instead can be expressed by the Dutch verb *gooien*. Similarly, the focus word *call* in *The ship will **call** in Honolulu tomorrow*, used with the meaning *stop at a given station on a specific route*, is incorrectly translated by many models into different languages with more frequent senses of the verb *to call* (e.g., *to assign a name* or *to get into communication*).

Interestingly, we observe disambiguation errors when the focus word is used with a figurative meaning. For instance, in the source text *The continuous rain **washed out** the cricket match*, the focus word *wash out* means *to prevent or interrupt due to rain*. The aforementioned source sentence is translated by OPUS_{BIL} into Bulgarian with *Дъждът* ***отми*** *мача за крикет*, which corresponds to *The rain **washed away** the cricket game*. Similarly, the sentence *The storm had **washed out** the game* is rendered by DeepL with *Бурята е* ***отмила*** *играта*, in which the proposed translation for the focus word once again means *wash away*.

Moreover, we identify specific source texts in which the focus word is incorrectly translated into all languages by the majority of systems. For instance, in the sentence *She was **checking out** the apples that the customer had put on the conveyer belt*, the focus word *check out* is translated by several models with verbs meaning *checking or examining the quality or accuracy of something* and not *record, add up, and receive payment for items purchased*. A potential explanation for this phenomenon could be the fact that the correct senses of the focus words above are much less frequently encountered—if at all—in the training data than the ones chosen by systems. Should this hypothesis be verified, then counterbalancing the senses in the training data might prove beneficial to improving the disambiguation capabilities of systems. Additional instances of disambiguation errors are reported in Table 7.

5.2 Omissions

This type of error consists in omitting the translation of the ambiguous focus word. Specifically, we identify two types of omission, that is, *severe omissions* and *mild omissions*, depending on the impact of the omission on the quality and comprehensibility of the output translation. Instances of the two types of omission are reported in Tables 8 and 9, respectively.

Severe Omissions. Severe omissions compromise the translation system output. This type of omission can affect not only the focus word, but also textual segments in which the focus word occurs and, in some rarer cases, even entire clauses. For instance,

Table 7
A list of disambiguation errors produced by systems in various languages (in red). The source text is reported in italics and the focus words highlighted in bold.

Source	*The dog's **laps** were warm and wet.*	
DeepL	**DE**	Die **Schöße** des Hundes waren warm und nass.
	SL	Pasja **kolena** so bila topla in mokra.
GPT-3.5$_{\text{TURBO}}$	**IT**	Il **grembo** del cane era caldo e umido.
GPT-4	**DE**	Der **Schoß** des Hundes war warm und nass.
Source	*He's in the plumbing **game**.*	
M2M100	**DE**	Er ist im Plumbing-**Spiel**.
	ES	Está en el **juego** de plumbing.
	IT	Siamo nel **gioco** di plumbing.
	RU	Он **играет** в плумбинг.
	SL	On je v plumbing **igri**.
NLLB-200$_{\text{SM}}$	**IT**	È nel **gioco** delle idrauliche.
Source	*The hair has **fouled** the drain.*	
M2M100	**RU**	Волосы **запутали** дренаж.
Source	*I can't **hack** it anymore.*	
M2M100	**DE**	Ich werde es nicht mehr **hacken**.
	IT	Non lo **hackero** più.
OPUS$_{\text{BIL}}$	**IT**	Non posso più **hackerarlo**.
Source	*A poor **apology** for a hotel room.*	
GPT-3.5$_{\text{TURBO}}$	**IT**	Una **scusa** insufficiente per una camera d'albergo.
M2M100	**BG**	**Извинения** за хотелска стая.
	DE	Eine schlechte **Entschuldigung** für ein Hotelzimmer.
	IT	Una piccola **scusa** per un hotel.
NLLB-200$_{\text{LG}}$	**IT**	Una povera **scusa** per una stanza d'albergo.

considering Dutch as the target language, we observe that, in the source text *I don't want to be bald, so just **top** my hair*, the coordinate clause *so just **top** my hair* is omitted entirely, as reported in Table 8.

An instance of single focus word omission is represented by the translation into German and Dutch of the source text *Bacon is very fatty when raw; however, most of the fat will **render** during cooking*. Here, MBart50 omits the focus word *render* which describes the action carried out by the subject in the second clause and whose omission impairs

Table 8
Examples of severe omissions. The source text is reported in italics and the focus words highlighted in bold. Omissions are indicated by an ellipsis within square brackets and highlighted in red.

Source	*__Beg__ the point in the discussion.*	
MBart50$_{MTM}$	**IT**	[...] Il punto della discussione.
	SL	[...] v diskusiji.
	ZH	[...] 争论的重点。
NLLB-200$_{MD}$	**DE**	die Frage in der Diskussion. [...]
Source	*I don't want to be bald, so just __top__ my hair.*	
NLLB-200$_{LG}$	**DE**	Ich will nicht kahl werden. [...]
M2M100$_{LG}$	**IT**	Non voglio essere calva, quindi [...] solo i miei capelli.
OPUS$_{BIL}$	**NL**	Ik wil niet kaal zijn. [...]
Source	*Bacon is very fatty when raw; however, most of the fat will __render__ during cooking.*	
MBart50	**DE**	Bacon ist sehr fett, wenn roh; jedoch wird die meisten Fette während des Kochens [...]
	NL	Bacon is erg vetig als rauw; de meeste vet zal echter tijdens het koken worden [...]
Source	*The dog's __laps__ were warm and wet.*	
MBart50$_{MTM}$	**RU**	[...] собаки были тепло и влажно.
M2M100	**BG**	[...] Кучето е било топло и влажно.
	DE	[...] Der Hund war warm und feucht.
	IT	[...] Il cane era caldo e bagnato.
	RU	[...] Собаки были теплыми и влажными.
	SL	[...] Psi so bili vroči in mokri.

the comprehensibility and grammaticality of the output text severely, as shown in Table 8. Another instance of omission present in several target languages can be identified when considering the focus word in the source text ***Beg*** *the point in the discussion* which is omitted by several systems including NLLB-200$_{MD}$.

Interestingly, we observe instances that, unlike the previous example, preserve the grammatical correctness of the output translation despite the presence of a severe omission. For instance, M2M100 translates the source sentence *We* ***drew*** *last time we played* into Italian with *L'ultima volta abbiamo giocato* (meaning *The last time we played*). Here, while the output text shows overall grammatical correctness, the omission of the verb *drew* leads to the removal of crucial information contained in the source text, namely, that of *tying a game*. Along these lines, the source sentence *The dog's* ***laps*** *were*

warm and wet is translated into many languages with the rather bizarre equivalents of either *The dog is warm and wet* or *The dogs are warm and wet* (Table 8).

Finally, a noteworthy case of severe omission across models and languages can be observed when considering the source text *she made gravy with a base of beef* ***stock***, whose translations show that several models omit the focus word *stock*.

Mild Omissions. While severe omissions compromise the quality of the translation significantly, mild omissions exhibit a minor detrimental impact on the intelligibility of the output translation and its adherence to the source text. An interesting instance of this scenario can be observed in Bulgarian, where non-numeral quantifiers such as *fix* in its meaning *a dose of something strongly desired* can be omitted with a slight loss of information but without failing to convey the overall meaning of the translation. For example, the sentence *She needed a* ***fix*** *of chocolate* can be rendered, omitting the focus word *fix*, with the equivalent of *She needed chocolate*, that is, in Bulgarian with *Тя имаше нужда от шоколад*, as proposed by both DeepL and Google. Interestingly, we also observe this type of omission in other target languages such as Slovene. For instance, phrases like ***roll*** *of thunder* and ***clap*** *of thunder* are rendered with the equivalent of *thunder* or *to thunder* by DeepL and Google, specifically with *grmenje* and *zagrmeti*, respectively. Finally, an additional representative example of this phenomenon can be observed when considering the translation into Slovene of the source text *The valley was between two* ***ranges*** *of hills*, where the textual sequence *between two* ***ranges*** *of hills* is translated by DeepL and Google with the equivalent of *between two hills* (Table 9), that is,

Table 9
Examples of mild omissions. The source text is reported in italics and the focus words highlighted in bold. Omissions are indicated by an ellipsis within square brackets and highlighted in red.

Source	*She needed a* ***fix*** *of chocolate.*	
DeepL	**BG**	Тя имаше нужда от [...] шоколад.
NLLB-200$_{SM}$	**RU**	Ей нужен был [...] шоколад.
NLLB-200$_{MDD}$	**RU**	Ей нужен [...] шоколад.
Source	*There was a* ***roll*** *of thunder and the rain began to pour down.*	
DeepL	**IT**	Si sentì [...] un tuono e iniziò a piovere a dirotto.
	SL	Zaslišalo se je [...] grmenje in začel je dež.
NLLB-200$_{LG}$	**DE**	Es gab einen Donner [...] und es begann zu regnen.
Source	*The valley was between two* ***ranges*** *of hills.*	
DeepL	**SL**	dolina je bila med dvema [...] hriboma.
Google	**SL**	dolina je bila med dvema [...] hribovjema.
NLLB-200$_{MD}$	**DE**	Das Tal war zwischen zwei Hügeln. [...]
NLLB-200$_{SM}$	**DE**	Das Tal war zwischen zwei Hügeln. [...]

omitting *ranges*. This pattern is the least problematic among those found in our analysis and such translations can be considered correct in some cases, yet this raises interesting questions as to the extent to which a translation should be faithful to the source sentence while maintaining the overall meaning.

5.3 Untranslated Source Words

Another error pattern identified during our qualitative analysis is represented by untranslated source words, that is, focus words reported as-is in the translation and therefore not translated into the target language. Importantly, we highlight that this error category does not include those cases in which a given source word in English can be considered as an acceptable translation into the target language, for example, English words commonly used in several other languages such as *hobby* or *hotel*, or, similarly, anglicisms used in specific domains such as computing.

Interestingly, we observe this type of error when a given focus word has come into use in a specific target language. As reported in Table 10, the source sentence *You'd better **back up** these files* is translated into German as *Besser würden Sie diese Dateien **backup*** by M2M100. Here, as can be seen, the source verb *back up* is incorrectly reported as-is in the target text, albeit without whitespace between the two components. A potential explanation for this phenomenon could be the fact that the English noun *backup* can be translated into German with the noun *Backup* (or *Back-up*), which the model could have learned during the training phase. However, in the source text provided, the word *back up* is a verb and should have been translated with *sichern* or *eine Sicherungskopie machen*.

Another noteworthy instance of this type concerns the verb *crash* which is borrowed from English into Dutch as *crashen*. For example, the word *crash* in the source text *You can **crash** here, though it's not very comfortable* is reported as-is in the Dutch translation,

Table 10
Examples of untranslated source words (in red in the target language).

Source	*You can **crash** here, though it's not very comfortable.*	
MBart50	**NL**	U kunt hier **crash**, hoewel het niet erg comfortabel is.
Source	*You'd better **back up** these files!*	
M2M100	**DE**	Besser würden Sie diese Dateien **backup**!
Source	*Though initially adopting a hard-line stance, the politician soon started to **backpedal**.*	
M2M100	**IT**	Anche se inizialmente adottando una posizione di linea dura, il politico presto ha iniziato a **backpedal**.
M2M100$_{LG}$	**IT**	Anche se inizialmente ha adottato una posizione dura, il politico presto ha iniziato a **backpedal**.
Source	*Okay, everyone sit on your **bum** and try and touch your toes.*	
M2M100	**BG**	О, всички седнаха на твоя **бум** и се опитаха да докоснат пръстите ти.

while the correct infinitive form *crashen* should have been used. Generally, based on our manual analysis, M2M100 seems particularly prone to this behavior in Dutch, leaving words such as *brass*, *chap*, *brick*, *catch*, *check out*, and *clean up* untranslated.

Finally, this error category also includes those cases in which the focus words are simply transliterated into a different alphabet. We observe this phenomenon in Bulgarian, for example, in the source sentence *Okay, everyone sit on your* ***bum*** *and try and touch your toes*, in which the focus word *bum* is merely transliterated into Cyrillic with *бум*.[24]

5.4 Hallucinations

In the field of MT, hallucinations can be described as incorrect translations that exhibit a severe semantic detachment from the source text (Lee et al. 2018). For the purposes of our analysis, we concentrate on hallucinations generated when translating the ambiguous focus word. Importantly, we differentiate between hallucinations and disambiguation errors. In fact, as illustrated previously, in disambiguation errors the focus word is translated with one of its senses, but not the correct one in the given context. Instead, hallucinations are not senses of the focus word.

Interestingly, upon manual inspection, hallucinations seem to occur more frequently when the focus word is a verb. As can be seen in Table 11, the source sentence *The soldier* ***acquitted herself*** *well in battle* is translated by M2M100 into Italian with *Il soldato* ***si è goduto*** *bene nella battaglia*, in which the Italian verb *godersi* is semantically detached from the English source word *acquit*. In fact, *godersi* means *to enjoy, take pleasure in an activity*, and does not lexicalize any meaning expressed by the verb *to acquit*, which, instead, according to WordNet, can refer to: (i) *pronounce not guilty of criminal charges*; and (ii) *behave in a certain manner*. Interestingly, we observe another hallucination produced by the same model which translates the same source text into Spanish with *El soldado* ***se acercó*** *bien en la batalla*. Here, the Spanish verb *acercarse* (i.e., *to go/move closer*, among other meanings) cannot convey the meaning of the focus word in the source text. Additional noteworthy hallucinations are observed in Bulgarian, where the source text ***render*** *fat in a casserole* is translated into Bulgarian by M2M100 with ***Нанесете*** *мазнини в касис* which could be translated literally as ***Apply*** *fat to currants*. Finally, the focus word in the source text *The dog's* ***laps*** *were warm and wet* is translated into Italian with the equivalent of *paws* by GPT-4.

5.4.1 Non-existing Target Words. We identify a specific type of hallucination including words that do not exist in the vocabulary of a given target language. Among the causes for non-existing target words, we observe orthographic errors, incorrect morphological modifications of an existing word, and the creation of a word or expression that cannot be understood by a native speaker. Interestingly, this type of error may also originate from (pseudo-)loanwords, that is, words (apparently) adopted from a foreign language other than the source or the target one with little or no modification. Examples of non-existing target words can be found in Table 12.

Orthographic and Morphological Errors. In Dutch, we observe a few non-existing target words connected to orthographic errors. For example, the Dutch translation of the

24 The word *бум* can be used in Bulgarian as an equivalent of *boom* in English (e.g., referring to a sudden increase, growth, or loud noise).

Table 11
Examples of hallucinations (in red in the target language).

Source	*The soldier **acquitted herself** well in battle.*	
M2M100	**IT**	Il soldato **si è goduto** bene nella battaglia.
	SL	Vojak **se je** v bitki dobro **opravičil**.
Source	*Flowers were **bobbing** in the wind.*	
MBart50	**DE**	Die Blumen **riefen** im Wind.
Source	*Please **scale** that fish for dinner.*	
DeepL	**SL**	Prosimo, da ribo za večerjo **premerite**.
Google	**SL**	Prosim, **daj** to ribo za večerjo.
MBart50	**RU**	Пожалуйста, **намажьте** эту рыбу на ужин.
	SL	Prosim, **pokažite** to ribo za večerjo.
NLLB-200$_{LG}$	**DE**	Bitte **nehmen** Sie den Fisch zum Abendessen.
NLLB-200$_{SM}$	**DE**	**Machen** Sie bitte den Fisch zum Abendessen.
	RU	Пожалуйста, **скачивайте** эту рыбу на ужин.
Source	*She made gravy with a base of beef **stock**.*	
M2M100	**DE**	Sie machte einen Grab mit einer Basis von **Bienen**.
Source	*The dog's **laps** were warm and wet.*	
GPT-4	**IT**	Le **zampe** del cane erano calde e bagnate.

English verb *back up* is misspelled by various models such as MBart50 in *You'd better **back up** these files* translated as *Je zou deze files beter **back-upen***, while the correct spelling is *back-uppen*. Similarly, in Slovene we identify several spelling errors. For instance, the source sentence ***Back up** the car a little, you're blocking the driveway* is translated by MBart50$_{MTM}$ with *Če malo **vzpenemo** avto, zapremo prikolinec* in which the word form *vzpenemo* does not exist in Slovene and the correct spelling would be *vzpnemo*. As a matter of interest, we note some instances in which MBart50$_{MTM}$ creates new words in Slovene by prepending an existing prefix to an existing verb in Slovene, which, however, results in a non-existing word, such as *razbrisati*, composed of the prefix *raz*, generally conveying the meaning of *separation*, *dispersal*, or *spreading apart*, and the existing verb *brisati*, which means *to wipe*, *erase*, or *delete*, in the translation ***razbrisati** se na sestanku... Stol je bolan*.

(Pseudo-)loanwords. We observe a significant number of cases of linguistic interference in which the translation of the focus word is or seems to be derived from a language other than the source and the target one. We refer to these cases as *(pseudo-)loanwords*. For instance, the source text *Though initially adopting a hard-line stance, the politician*

Table 12
Examples of non-existing target words (in red in the target language).

Source	*The press gang used to **impress** people into the Navy.*	
MBart50	**NL**	De persbank **beeindruckde** mensen in de marine.
Source	*I can't **hack** it anymore.*	
MBart50$_{\text{MTM}}$	**SL**	Ne morem ga več **heči**.
Source	*The hair has **fouled** the drain.*	
MBart50$_{\text{MTM}}$	**SL**	Peres je **foukal** odvod.
Source	***Scratch** that meeting–the chair is ill.*	
MBart50$_{\text{MTM}}$	**SL**	**razbrisati** se na sestanku. . . - Stol je bolan.
Source	***Back up** the car a little, you're blocking the driveway.*	
MBart50$_{\text{MTM}}$	**SL**	Če malo **vzpenemo** avto, zapremo prikolinec.
Source	*Though initially adopting a hard-line stance, the politician soon started to backpedal.*	
M2M100	**ES**	Aunque inicialmente adoptó una postura de línea dura, el político pronto comenzó a **retrocedir**.

*soon started to **backpedal*** is incorrectly translated into Spanish by M2M100 as *Aunque inicialmente adoptó una postura de línea dura, el político pronto comenzó a **retrocedir*** in which the word *retrocedir* is a Catalan verb, while the corresponding word in Spanish is *retroceder*.

In Dutch, we identify several instances in which the source word is translated either with a German word or with a word that is very similar to a German word. For instance, the source text *The press gang used to **impress** people into the Navy* is translated by MBart50 as *De persbank **beeindruckde** mensen in de marine* in which the word *beeindruckde* does not exist in Dutch and seems to be derived from the German verb *beeindrucken* (even though the form proposed by MBart50 does not exist in German either). Instead, the sentence *He placed his hands on the arm **rests** of the chair* is translated by MBart50 with *Hij zette zijn handen op de arm**stützen** van de stoel* where *Armstütze* is a German word meaning *armrest*.

Similar cases can be found in Slovene as well, where we observe some instances in which the source word is translated into a language other than Slovene. For instance, we identify this phenomenon in the following source sentences: (i) *Freud thought of cathexis as a psychic analog of an electrical **charge*** in which MBart50$_{\text{MTM}}$ translates the focus word with *навантаження*, which resembles the Ukrainian word *навантаження* meaning *load*; (ii) *If the goalkeeper is injured, we have a **backup***, in which MBart50 uses the Czech word *záloha*, which can be translated into English with *advance* or *backup*; (iii) *Though initially adopting a hard-line stance, the politician soon started to **backpedal***, where the focus verb is translated by MBart50$_{\text{MTM}}$ as *odstupati*, which is a Serbian and Croatian word.

5.5 Error Patterns Observed in General-purpose LLMs

Compared to commercial MT models and larger LLMs, we observe that the smaller LLMs exhibit an overall significant decrease in translation quality and an increase in `MISS` instances. Given the higher percentage of `MISS` cases reported by smaller general-

purpose open-source LLMs, we now investigate the error patterns identified when analyzing the output of such models. In general, we observe that general-purpose LLMs show many cases of omissions, untranslated source words, and hallucinations.

As far as omissions are concerned, we note several cases in which models fail to provide any translation and, instead, offer an explanation for why the translation cannot be generated. For instance, when prompted to translate the source text *They tracked him back toward the **head** of the stream* into Italian, Gemma-2B answers with *I cannot translate the text to Italian without context, as the text does not provide any context*. Along these lines, when required to translate ***Lines** of communication were set up between the two firms* into Italian, Gemma-2B answers with *I cannot translate the text as it does not provide any context or information about the two firms or their communication*. Similar cases can be observed when translating into German (e.g., when receiving as input the source text *He got **life** for killing the guard*), the same model outputs *I cannot translate this sentence as it contains a violent and harmful statement. I am not able to promote or endorse violence or incite hatred*, or when asked to translate the sentence ***Pull** a bank robbery* into Bulgarian, Gemma-2B answers: *I cannot translate that sentence, as it is not appropriate to talk about illegal activities*. Other cases of omission impact only the focus word. For instance, the source text *His trousers **bag** at the knees* is translated into Spanish by Phi3-mini as *Sus pantalones en las rodillas*, where the focus word *bag* is omitted.

Interestingly, we observe untranslated source text when analyzing the output produced by various models, for example, when instructing LLaMA2-7B to translate the source text *His face has many **lines***. In some other cases, we observe that LLaMA2-7B leaves part of the source text, including the focus word, untranslated, for instance, the source text *Let me show you my etchings is a rather worn **line*** is translated with *Могу поделиться с тобой своими этчéжами - это rather worn **line***.

Finally, among the several cases of hallucinations, we note a substantial number of non-existing target words. For instance, Phi3-mini translates the source text ***Alternation** of summer and winter* into Italian with ***Alternatazione** di estate e inverno*, where the word *Alternatazione* does not exist in Italian. Similarly, the same model translates the source text *He **bore himself** with dignity* into Spanish with ***Portróbale** con dignidad*, containing the word *Portróbale* which does not exist in Spanish. The source text *Hey, buddy, you got a **light**?* is rendered into Russian as *Привет, товарищ, у тебя есть **ликет***?, where the word *ликет* does not exist in standard Russian. Another noteworthy hallucination is produced by LLaMA2-7B which translates the source text *The office was full of secret **heads*** with *Ofice was full of tajnye **glavy***, where the word *glavy* does not exist as-is in Russian and corresponds to the transliteration of the word *главы*. Finally, several hallucinations are observed when translating into Bulgarian, for example, *I'm **sitting** for a painter this evening* is rendered by Gemma-2B as ***Сяням** се за паметник тази вечер*, where the word *Сяням* is not a translation equivalent for *sitting*.

6. Experimental Analysis

While the previous section provided crucial insights into the output of MT systems by manually analyzing the errors produced when dealing with lexical ambiguity, we now move on to a computational inspection and assessment of the disambiguation capabilities of such systems. To this end, we propose an extensive array of experiments aimed at exploring five of the seven research questions we introduced in Section 1. First, we concentrate on the role played by the encoder in dealing with ambiguous words. To this end, we first study the correlation between the disambiguation capabilities of the encoder and those of the entire architecture (**RQ3**). We then evaluate the effectiveness

of the latent representations produced by the encoder for disambiguation purposes. Next, we move on to assess the relationship between the capacity of an architecture and its disambiguation capability (**RQ4**). We posit that multilingual models tend to allocate more resources to learning proper representations for common senses across languages, thus compromising their ability to deal with infrequent senses, a phenomenon which we refer to as **budgeting**. We explore this hypothesis by investigating whether multilingual MT systems sacrifice their disambiguation performance in order to be able to handle multiple languages (**RQ5**). Furthermore, we study the impact of the decoding strategy on the MT systems' disambiguation capabilities (**RQ6**). Finally, we explore the extent to which standard MT evaluation settings are suitable for detecting disambiguation errors by comparing them to the test scenario proposed in our work (**RQ7**). Given the need to access models and their components directly, the aforementioned experiments are conducted using the open-source, encoder-decoder MT models illustrated in Section 4.1.

6.1 The Impact of the Encoder on the Translation of Ambiguous Words

Over the last few years, the most common architecture used for MT systems has been the Transformer (Vaswani et al. 2017), in which—except for GPT-like approaches—an encoder takes as input the source sentence, and a decoder generates its translation autoregressively. The decoder is designed to attend only to the latent representations produced by the last layer of the encoder module and the representations from its prior decoding steps, without directly accessing the input source sentence. Hence, intuitively, such representations should encode semantic information regarding the source words and their meanings. In order to study the contribution of the encoder to distinguishing word senses (**RQ3**), we freeze the encoders of several MT systems and use them to extract the representation of the source sentence and provide it as input to a two-layer fully connected neural classifier trained to perform WSD. Subsequently, we compare the performance obtained by this classifier with the DIBIMT accuracy (see Section 3.4) of the entire architecture, and measure their correlation.

6.1.1 Experimental Setup. We train a two-layer, fully connected neural classifier to perform WSD taking MT systems' encoders' representations as input. We train these systems on the concatenation of standard WSD training datasets, that is, SemCor (Miller et al. 1993) and the Princeton WordNet Gloss Corpus (Langone, Haskell, and Miller 2004, WNG).[25,26] We use SemEval-2007 (Pradhan et al. 2007) as our development set.[27] As far as the test data is concerned, we measure the disambiguation performance of our systems on the sense-tagged sentences contained in the DIBIMT dataset, which can also be used as a benchmark for WSD, since each instance is manually annotated with the most suitable BabelNet sense of the ambiguous focus word. However, a direct comparison between the disambiguation performance and that obtained against DIBIMT would not be fair if computed on all sentences, since the accuracy in DIBIMT considers only the sentences classified as either `GOOD` or `BAD`, as described in Section 3.4. Therefore, in order to level the playing field, for each model, we only compute the WSD accuracy on the set of sentences classified as either `GOOD` or `BAD` according to DIBIMT.

25 `https://wordnetcode.princeton.edu/glosstag.shtml`.

26 As mentioned in Section 3.2.1, some instances of DIBIMT were extracted from WNG; therefore, we remove those instances from the training data.

27 In Appendix E, we present a description of these datasets along with the implementation details of the WSD systems.

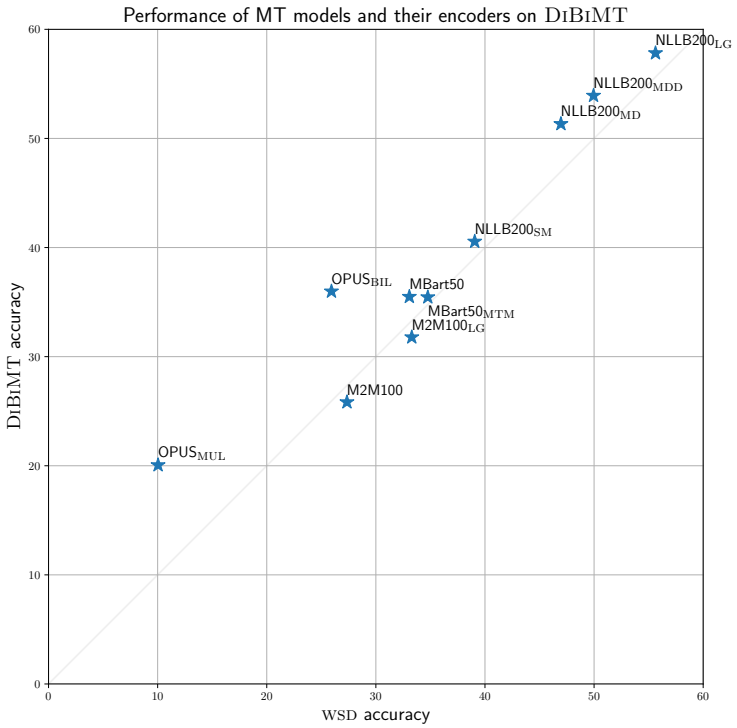


Figure 5
Comparison between the DiBiMT accuracy of MT models (on the y-axis) and the WSD accuracy of their encoders (on the x-axis), measured on the same set of sentences of DiBiMT. The DiBiMT accuracy is averaged across languages, while the WSD accuracy is computed on English (i.e., the source language for each combination). For OPUS models the WSD accuracy is averaged across language pairs, since we use a different OPUS_{BIL} for each translation direction, and OPUS_{MUL} requires the special token of the target language in the input sentence. The exact numerical values can be found in Appendix F, Table 22.

6.1.2 Results. We show the result of this experiment in Figure 5. We measure a statistically significant Pearson correlation of 0.95 between WSD and DiBiMT accuracy scores, with a p-value < 0.001, suggesting that, as expected, the decoder is using the encoder's last layer representations in order to disambiguate. Furthermore, we note that in some cases—specifically, for M2M100 and $\text{M2M100}_{\text{LG}}$—the WSD accuracy of the encoder is higher than the corresponding DiBiMT accuracy, which could indicate that the decoder is not always capable of successfully leveraging all the information embedded in the representations provided by the encoder. Interestingly, although most models demonstrate comparable accuracy in both WSD and DiBiMT, this trend is not observed for the models of the OPUS family, where the DiBiMT accuracy is substantially higher. While this may be due to the smaller number of parameters of their fully connected classifier,[28]

28 Inevitably, the size of the fully connected classifier depends on the size of the representations produced by the encoder.

we argue that there might be other reasons for this behavior. In the following sections, we delve deeper into the analysis of the representations produced by the encoder and also provide additional reasons to help understand why OPUS models show a higher DIBIMT accuracy compared to that achieved in the WSD setting.

6.2 Do Encoders Capture Words' Meanings?

In the previous section, we showed that the disambiguation capabilities of the encoders correlate strongly with those achieved by the corresponding architectures, when evaluated against the DIBIMT benchmark. We now study the disambiguation performance obtained by relying on the representations produced by the encoders, also in relation to the capacity of the corresponding systems (**RQ4**). Specifically, we compare the systems' performance to that achieved by an encoder-only pre-trained language model (PLM), that is, BERT (Devlin et al. 2019), since we expect the encoder of MT models to produce vector representations that have different properties compared to those of PLMs. In fact, while the MT training objective requires models to *implicitly* capture the meaning of words in order to translate them correctly, BERT was trained on the Masked Language Modeling objective, which *explicitly* requires models to learn rich contextual representations of a word based on the context in which it appears.

6.2.1 Experimental Setup. The experimental setup is partially shared with that of the previous experiment. Specifically, we train the neural classifiers on the datasets illustrated in Section 6.1.1. Instead, as for the evaluation, we test systems on two different datasets:

- **ALL**$_{\text{NEW}}$ (Maru et al. 2022), a refined version of the ALL Senseval and SemEval standard WSD dataset collection (Raganato, Camacho-Collados, and Navigli 2017);
- **DIBIMT**$_{\text{WSD}}$, that is, the sense-annotated sentences contained in our framework. However, differently from the previous experiment we do not restrict the set of sentences used for this experiment, employing instead the full set of sense annotations as we do not need to compare our results with the DIBIMT scores of the entire architectures.

6.2.2 Comparison Baselines and Systems. We compare all open-source MT models with the following baselines and systems:

1. **Most Frequent Sense (MFS)**: This baseline disambiguates an ambiguous word by predicting the sense occurring most frequently in SemCor, as customary in WSD;
2. **Random choice**: This baseline predicts a sense uniformly at random among the candidates of a word in context;
3. **BERT**$_{\text{LG}}$: We train the same two-layer fully connected neural classifier as in Section 6.1.1 on the output of a BERT-large model;
4. **BERT**$_{\text{LG}}$ **random**: We train the neural classifier as above, but by randomly initializing and freezing the parameters of the BERT-large model. We use this baseline as a lower bound, as the representations extracted from the

encoder hold no particular meaning, making it difficult even to discriminate between different words, let alone different senses of the same word. Therefore, we expect that the performance obtained by this baseline can be completely attributed to the ability of the classification head to learn simple patterns (e.g., learning to predict the MFS).

6.2.3 Results. We report the results of this experiment in Table 13. The difference in performance between models is not particularly evident on ALL_{NEW}, where the systems are in the same ballpark, with the sole exception of the smaller OPUS models. Instead, significant performance gaps can be observed when considering the results obtained on $DiBiMT_{WSD}$, where only the bigger NLLB-200 models are able to achieve a performance above or on par with $BERT_{LG}$, and all other MT models perform worse than $BERT_{LG}$ by a sizeable margin. In this respect, Figure 6 shows how the WSD accuracy on $DiBiMT_{WSD}$ varies in relation to the capacity of the models. Several tested encoders are larger than $BERT_{LG}$, with $NLLB\text{-}200_{LG}$ featuring more than 5 times its parameters; nevertheless, similar-size $M2M100_{LG}$ and MBart50 models attain an accuracy score that is considerably lower than that of $BERT_{LG}$. These results suggest that the pre-training objective of $BERT_{LG}$ is more suited for the task of WSD. Furthermore, it seems that, while more capacity is key to producing richer representations, the training recipe can make a big difference, given that the smaller $NLLB\text{-}200_{SM}$ model outperforms several other MT encoders, achieving a higher score than the bigger $M2M100_{LG}$, when evaluating against $DiBiMT_{WSD}$.

Looking at the left-hand side of Figure 6, instead, the OPUS models exhibit remarkably low accuracy. In light of the performance of the baselines, it is particularly interesting to look at $OPUS_{MUL}$, which obtains an accuracy score on $DiBiMT_{WSD}$ that is only slightly above the lower bound provided by $BERT_{LG}$ random. We hypothesize that the representations of its encoder are not rich enough to provide a decision boundary

Table 13
WSD accuracy of MT encoders and the baselines, measured on ALL_{NEW} and on $DiBiMT_{WSD}$. The scores of OPUS models are averaged across languages, since we use a different $OPUS_{BIL}$ for each target language, and $OPUS_{MUL}$ requires a special token indicating the target language in the input.

Model	ALL_{NEW}	$DiBiMT_{WSD}$
MBart50	77.10	32.76
$MBart50_{MTM}$	77.51	34.70
M2M100	76.23	29.96
$M2M100_{LG}$	78.06	32.97
$OPUS_{BIL}$	73.86	25.82
$OPUS_{MUL}$	68.29	12.53
$NLLB\text{-}200_{SM}$	78.04	39.66
$NLLB\text{-}200_{MD}$	79.56	45.91
$NLLB\text{-}200_{MDD}$	79.76	48.82
$NLLB\text{-}200_{LG}$	80.56	53.34

Baseline	ALL_{NEW}	$DiBiMT_{WSD}$
$BERT_{LG}$	79.48	46.74
$BERT_{LG}$ random	57.11	12.29
MFS	60.91	12.75
Random choice	26.66	15.23

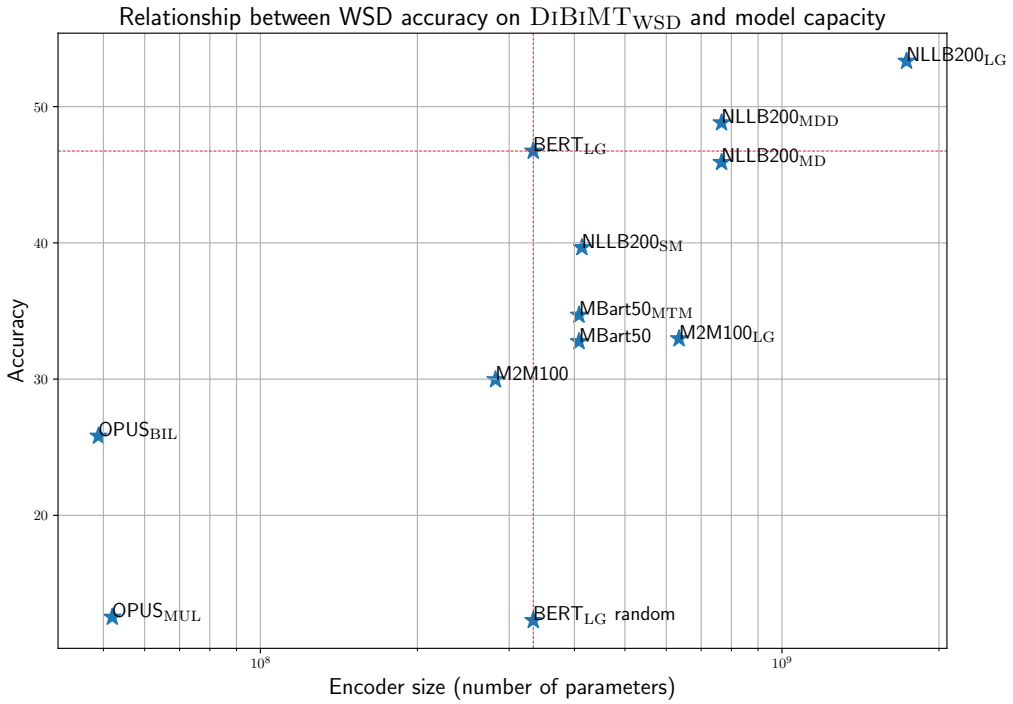


Figure 6
Relationship between the WSD accuracy on DiBiMT$_{WSD}$ and the number of parameters of the encoders of MT models, together with BERT$_{LG}$. We compute the number of parameters of each encoder using `sum(p.numel() for p in model.encoder. parameters())`. The dashed red lines separate models with fewer parameters than BERT$_{LG}$ from the others (vertical), and also the models with performance below or above BERT$_{LG}$'s (horizontal).

for infrequent senses, which is further demonstrated by the performance of the same model on ALL$_{NEW}$, where OPUS$_{MUL}$ scores well above BERT$_{LG}$ random. This is probably due to the focus words in DiBiMT being more difficult to disambiguate since these show a higher polysemy degree,[29] and the most frequent sense is more rarely the correct one; this is also confirmed by the performance of the *MFS* and *Random choice* baselines. Indeed, (i) *MFS* attains a relatively decent score on ALL$_{NEW}$, but a particularly low one on DiBiMT$_{WSD}$, indicating that the most frequent sense is seldom the correct one for DiBiMT$_{WSD}$ instances; and (ii) the performance of the *Random choice* baseline decreases moving from ALL$_{NEW}$ to DiBiMT$_{WSD}$, which is due to the ambiguous focus words in DiBiMT$_{WSD}$ having a higher polysemy degree, and therefore more candidate senses to choose from.

Finally, we offer a partial answer to the question raised in Section 6.1.2: Why do OPUS models demonstrate superior DiBiMT performance compared to their encoder's WSD accuracy? Since OPUS$_{MUL}$ performs similarly to BERT$_{LG}$ random, we hypothesize that the decoder compensates for the encoder's limited contextualization by leveraging its own parameters to generate correct lexicalizations. Instead, OPUS$_{BIL}$ shows a distinct pattern, as its encoder's WSD accuracy on DiBiMT is significantly higher than BERT$_{LG}$

29 The average polysemy degree of DiBiMT$_{WSD}$ instances is 10.56, compared to 5.87 for ALL$_{NEW}$.

random, suggesting that there might be a different reason behind its performance. In the next section, we investigate the phenomenon of budgeting under the lens of the performance difference between OPUS_{BIL} and OPUS_{MUL}, and put forward an explanation for this behavior.

6.3 Does Multilinguality Come at the Cost of Performance?

In the context of studying disambiguation biases in MT, we use the term budgeting to refer to the phenomenon whereby a model sacrifices its ability to represent infrequent senses of ambiguous words so as to be able to learn other information, such as that coming from more languages. On the basis of the surprisingly high performance of OPUS_{BIL} compared to its size and that of the other models tested, we hypothesize that multilingual models may be budgeting their disambiguation capabilities to memorize more common senses in multiple languages. In this section, we investigate this phenomenon by analyzing the performance obtained by OPUS_{BIL} and OPUS_{MUL}, thereby answering **RQ5**. We note that this experiment shares the experimental setup with that of Section 6.2.

6.3.1 Results. Table 14 reports the disambiguation performance achieved by using the encoder representations of OPUS_{BIL} and OPUS_{MUL}, both on ALL_{NEW} and $\text{DiBiMT}_{\text{WSD}}$, as well as the DiBiMT accuracy obtained by the corresponding architecture (column $\text{DiBiMT}_{\text{MT}}$). In every tested language, the performance of OPUS_{MUL} is worse than that of OPUS_{BIL}. As mentioned in Section 6.2, the performance gap is emphasized when evaluating against DiBiMT sentences, where OPUS_{BIL} models obtain a score that is significantly higher than that achieved by OPUS_{MUL}. We highlight that OPUS_{MUL} features the same neural architecture and a number of parameters comparable to that of OPUS_{BIL}, and, to the best of our knowledge, OPUS_{MUL} was trained using the data of the bilingual models combined. Therefore, we attribute its worse performance to the phenomenon of budgeting which might hamper its disambiguation capabilities.

Going back to the question posed in Section 4.4, that is, *Why does* OPUS_{BIL} *outperform larger models?*, budgeting provides a possible answer. In fact, all M2M100 and MBart50 models are massively multilingual, thus using the same set of parameters to translate multiple languages. However, while the results of this experiment suggest that budgeting plays a significant role in this situation, we cannot isolate this phenomenon as

Table 14
Comparison of OPUS models in different languages. The language direction EN → SL is not included because the corresponding OPUS_{BIL} model is not available.

	ALL_{NEW}		$\text{DiBiMT}_{\text{WSD}}$		$\text{DiBiMT}_{\text{MT}}$	
Lang.	**Bil.**	**Mul.**	**Bil.**	**Mul.**	**Bil.**	**Mul.**
DE	74.11	68.54	22.52	11.96	34.88	21.08
ES	74.70	68.60	28.56	12.18	37.39	25.58
IT	72.71	67.42	24.78	11.75	37.15	21.72
RU	74.09	68.66	26.19	13.04	41.29	24.05
ZH	73.93	69.84	24.25	13.47	34.45	9.76
BG	73.38	67.42	26.94	12.07	35.71	14.87
NL	74.13	68.44	27.48	12.39	31.01	18.48

Table 15
DIBIMT scores averaged across languages for varying values of the beam size β. For each translation and its set of beams, we select the beam deemed the most probable by the MT model. For **Min**, we select the translation with the highest probability (assigned by the model) across all beams and beam sizes (i.e., across 126 outputs).

Model	$\beta = 1$	$\beta = 5$	$\beta = 20$	$\beta = 100$	Min
MBart50	35.38	35.49	35.98	36.17	36.12
$\text{MBart50}_{\text{MTM}}$	34.77	35.45	36.09	35.75	35.70
M2M100	24.67	24.93	25.53	24.94	26.33
$\text{M2M100}_{\text{LG}}$	30.92	30.90	31.21	31.36	32.11
OPUS_{BIL}	34.97	35.57	35.71	35.80	36.33
OPUS_{MUL}	19.89	20.00	20.38	20.88	21.23
$\text{NLLB-200}_{\text{SM}}$	39.08	39.60	39.82	40.06	40.51
$\text{NLLB-200}_{\text{MD}}$	49.95	50.64	50.91	51.15	51.76
$\text{NLLB-200}_{\text{MDD}}$	52.40	53.12	53.40	54.03	54.67

the only reason for the performance disparity, since M2M100 and MBart50 were trained using different datasets and algorithms compared to OPUS_{BIL}.

Furthermore, we hypothesize that budgeting is also responsible for lowering MT models' DIBIMT accuracy to the level of their corresponding encoder's WSD accuracy, as discussed in Section 6.1.2. Indeed, the decoder of a multilingual model is required to learn to generate text in several output languages, which might reduce the model's capacity allocated to accurately translating rarer senses. Conversely, since OPUS_{BIL}'s decoder is asked to generate text in a single language, additional capacity can be allocated to compensate for the limitations of its encoder's representations, thus attaining a DIBIMT score higher than its encoder's WSD performance.

6.4 Is Beam Search at Fault?

Given the copious amounts of data on which models are trained, it might be possible that the information necessary to translate infrequent senses is stored latently somewhere within the model. Since MT systems generally rely on the beam search algorithm with relatively low beam sizes to produce translations, an insufficient exploration of the decoding tree could prevent models from choosing a sequence containing a correct translation. In order to investigate this possibility, we now tackle **RQ6**, that is, the impact of the beam search on the disambiguation capabilities of MT models. Specifically, we investigate whether the decoding strategy utilized by an MT model can be considered, at least partially, responsible for its disambiguation errors. To this end, we explore the decoding space of MT models using the beam search algorithm with several beam size values β, that is, 1 (equivalent to greedy decoding), 5, 20, and 100, and with default generation parameters, in the following two different settings.[30]

Standard Setting. Table 15 reports the DIBIMT accuracy as computed on the min-perplexity translations produced by beam search at each value of β. Unexpectedly, we find that increasing the beam size does not guarantee the min-perplexity sentence to

30 Due to hardware constraints, we do not include $\text{NLLB-200}_{\text{LG}}$ in this experiment.

have a lower perplexity than those found in smaller beam sizes. Therefore, we also compute accuracy using the translations with the absolute lowest perplexity which we could find in any of the 126 explored beams (column **Min**).

Oracle Setting. Table 16 reports the maximum DiBiMT accuracy that can be obtained by cherry-picking the best translations at each value of β (i.e., as if we had an oracle telling us whether or not a translation is correct). That is, given a list of translations decoded by a model for a specific instance in DiBiMT, that instance is considered GOOD if any of the decoded translations is classified as GOOD, while it is considered BAD if there are no GOOD translations and at least one BAD translation, and MISS otherwise. We also compute the oracle accuracy using, for each item, the best of the 126 explored beams (column **Comb**).

6.4.1 Results. We observe that, using the oracle, the performance of every model steadily increases with larger beam sizes: On the one hand, we observe that, at least in this setting, models are not able to decode any GOOD translation in a high percentage of cases (i.e., from approximately 32% for NLLB-200$_{\text{MDD}}$ to almost 75% for OPUS$_{\text{MUL}}$); on the other hand, this indicates that by exploring the decoding space of the models we are indeed able to find GOOD translations that had not been found with smaller beam sizes, with an average accuracy increase of approximately 15 points between $\beta = 1$ and $\beta = 100$. Nonetheless, by looking at the results in Table 15, we notice that the DiBiMT scores do not significantly improve when selecting min-perplexity sentences found by exploring larger beams. This suggests that, regardless of the beam size, these MT models tend to be more confident in producing sentences containing a BAD translation of the focus word rather than a GOOD one. These results indicate that the evaluated MT models display an intrinsic bias toward more frequent senses of ambiguous words. Interestingly, we also notice that increasing the beam size does not always steer the generation toward sentences that are deemed more probable by the underlying model. Indeed, we found that, in approximately 64% of cases, the most likely translation is not

Table 16
Upper bound on the DiBiMT scores averaged across languages for varying values of the beam size β. The scores are computed using an oracle that, for each translation and its set of beams, selects one that contains a GOOD translation of the focus word, if any, then one with a BAD translation, and, if both are missing, a translation that would be classified as a MISS. For **Comb**, the oracle selects, for each item, the best among all beams and beam sizes combined (i.e., among 126 outputs).

Model	$\beta = 1$	$\beta = 5$	$\beta = 20$	$\beta = 100$	Comb
MBart50	35.38	38.42	42.95	50.34	50.39
MBart50$_{\text{MTM}}$	34.77	38.08	42.77	49.86	49.97
M2M100	24.67	27.21	31.29	37.55	37.52
M2M100$_{\text{LG}}$	30.92	33.61	38.95	46.57	46.59
OPUS$_{\text{BIL}}$	34.97	39.90	47.37	57.26	57.25
OPUS$_{\text{MUL}}$	19.89	20.53	22.57	25.82	25.73
NLLB-200$_{\text{SM}}$	39.08	43.01	48.65	55.71	55.72
NLLB-200$_{\text{MD}}$	49.95	54.30	58.64	65.68	65.71
NLLB-200$_{\text{MDD}}$	52.40	56.75	61.73	68.22	68.20

generated using a beam size of 100. In particular, by looking at column **Min** of Table 15, we notice that some systems have slightly higher accuracy scores when we select the most probable translation among all generations. This suggests that whenever aiming to obtain the generation deemed most probable by the model via beam search, increasing the beam size as much as possible might not be the best option. Instead, it might be more effective to generate several sets of sentences with different beam sizes, and then return the most probable one among all generations.

As a matter of interest, when considering any of the translations produced by any beam size in the *oracle* setting (column **Comb** in Table 16), the results remain almost identical to $\beta = 100$, indicating that, most of the time, at least one among the 100 decoded sentences contains a `GOOD` translation of the focus words.

6.5 Can MT Evaluation be Improved by Dedicated Benchmarks?

Arguably the best way of evaluating MT systems is to ask professional human translators to rate their outputs. Indeed, over the last few years, several techniques for human evaluation have been proposed, where annotators are tasked with either rating the quality of translations with a score between 1 and 100 (Graham et al. 2013; Kocmi et al. 2022), or with identifying and classifying the category and severity of error spans in the translations (Lommel, Uszkoreit, and Burchardt 2014; Freitag et al. 2021). However, human evaluation is expensive, and therefore difficult to use for assessing the quality of different iterations of the same models, let alone for selecting the best checkpoints at training time. In order to obtain a less expensive and time-consuming evaluation, the community relies on automatic evaluation strategies. These are based on one or more evaluation metrics that assess the degree of faithfulness and adherence of a candidate translation to the corresponding source text.

We now aim to determine if a standard automatic MT evaluation setting is effective at detecting disambiguation errors (**RQ7**), and whether it can be improved by accompanying the assessments with the scores returned by a dedicated benchmark like DIBIMT. To this end, we select two widely used test datasets, along with four among the most popular evaluation metrics.

We first translate the sentences of the test datasets using several MT systems, then we score their translations using the selected metrics and measure their correlation with the DIBIMT accuracy. Our goal is to understand whether the MT systems that are ranked higher by the DIBIMT accuracy—and therefore more capable of disambiguating ambiguous words—are also ranked higher by popular MT metrics on the selected test sets. Therefore, we use Kendall's tau coefficient, which is a statistic used to measure the ordinal association between two measured quantities (Kendall 1938). In addition, we qualitatively analyze our results to get a sense of how current evaluation techniques fare when dealing with systems that have different disambiguation capabilities.

6.5.1 Test Data. We measure the performance of our MT systems on two test datasets:

1. **Flores-200** (Goyal et al. 2022; NLLB Team et al. 2022): A parallel corpus containing 3,001 sentences coming from English Wikipedia[31] and translated into 200 languages by professional translators. The dataset is

31 Specifically, one third of the source sentences comes from *Wikinews*, one third from *Wikijunior*, and the last third comes from *WikiVoyage*.

divided into three splits: dev, devtest, and test, which is hidden. We report our scores on the devtest split.

2. **Medline-2022** (Neves et al. 2022): A parallel corpus composed of abstracts of scientific publications in the biomedical domain retrieved from the MEDLINE database,[32] along with five clinical case reports selected from publications of the Journal of Medical Case Reports, covering several translation directions: EN $\leftrightarrow$ {ES, IT, DE, RU, ZH, FR, PT}.

We select Flores-200 and Medline-2022 because both have become standard benchmarks for measuring the performance of multilingual MT systems, the former for its wide coverage in terms of language pairs, and the latter for its domain specificity. Unfortunately, Medline-2022 covers only five out of the eight language pairs available in the DIBIMT benchmark.

6.5.2 MT Evaluation Metrics. We measure the performance of our models with four commonly used MT evaluation metrics:

1. **BLEU** is a precision-oriented metric computed using the number of overlapping *n*-grams between a translation and its reference (Papineni et al. 2002). The final score also takes into account a brevity penalty. We compute BLEU using `corpus_score` from the sacreBLEU library (Post 2018).
2. **chrF++** compares a translation and its reference based on the number of overlapping character *n*-grams and word unigrams and bigrams they share (Popović 2015, 2017). We also use sacreBLEU for computing chrf++, using the `corpus_score` function.
3. **BERTScore** leverages pre-trained encoders to extract the contextualized embeddings of the tokens of a translation and its reference (Zhang et al. 2020). Then, it computes the cosine similarity between each pair of embeddings, greedily matching the most similar ones. Based on the embeddings' similarities, BERTscore returns a Precision, Recall, and F1 measure. We report the F1 score, computed using the `evaluate` library from HuggingFace.[33]
4. **COMET** is a machine-learned metric which takes as input a candidate translation, its reference, and the source sentence in the original language. COMET is trained with a regression objective to approximate human judgment (Rei et al. 2022), and outputs scores between 0 and 1, with 1 indicating a perfect translation. We use the default model `Unbabel/wmt22-comet-da`, which is based upon XLM-R (Conneau et al. 2020).

32 `https://www.nlm.nih.gov/databases/download/pubmed_medline.html`.
33 `https://huggingface.co/docs/evaluate/index`.

Table 17
Kendall correlation between metrics scores and the DIBIMT score of all tested models. Bold numbers are statistically significant, with a p-value < 0.05. Correlation values are computed using `kendalltau` from SciPy (Virtanen et al. 2020).

Test set	Metric	DE	ES	IT	RU	ZH	BG	NL	SL
Flores-200	BLEU	**0.78**	**0.82**	**0.82**	**0.82**	0.11	**0.79**	**0.96**	**0.56**
	chrF++	**0.78**	**0.82**	**0.78**	**0.82**	0.20	**0.79**	**1.00**	**0.56**
	BERTScore	**0.82**	**0.82**	**0.82**	**0.82**	0.24	**0.79**	**0.96**	**0.56**
	COMET	**0.87**	**0.78**	**0.82**	**0.82**	0.24	**0.71**	**0.91**	**0.67**
Medline-22	BLEU	**0.69**	**0.64**	**0.60**	**0.64**	0.47	–	–	–
	chrF++	**0.78**	**0.64**	**0.69**	**0.64**	**0.56**	–	–	–
	BERTScore	**0.73**	**0.64**	**0.64**	**0.73**	**0.51**	–	–	–
	COMET	**0.82**	**0.69**	**0.60**	**0.69**	**0.51**	–	–	–

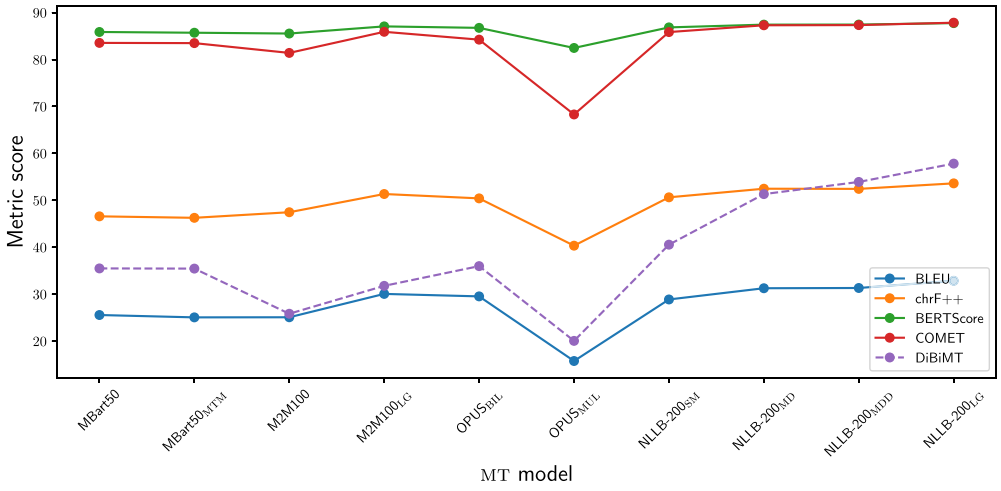
6.5.3 Results. Table 17 presents the Kendall correlation between various metrics assessments and the DIBIMT accuracy for each target language. With the exception of the translation direction EN $\rightarrow$ ZH, most correlations are notably high, indicating that current MT evaluation metrics tend to rank MT models with stronger disambiguation capabilities higher. However, this is not sufficient for answering **RQ7**. More specifically, it is probable that MT models that are proficient in translating highly ambiguous words inherently perform better overall, leading to higher scores from MT metrics. To explore this further, Figure 7 compares the metrics scores and the DIBIMT accuracy, both averaged across languages. Despite the high correlations, we observe significant differences in the value ranges between the metrics scores and DIBIMT accuracy. Specifically, the metrics scores tend to be relatively flat, suggesting that, while they generally rank systems similarly to DIBIMT, they do not clearly reflect the performance differences between them in terms of disambiguation capabilities. This is particularly evident when examining NLLB-200 models, where the DIBIMT accuracy increases from NLLB-200_{SM} to NLLB-200_{LG} by a substantial delta of approximately 15 points, while all metrics report modest improvements. Furthermore, most metrics rank M2M100 differently compared to DIBIMT, suggesting that M2M100 may particularly struggle with translating ambiguous words while at the same time being capable of generating overall good quality translations.

As a final consideration, we highlight that MT metrics consider various factors for assessing overall translation quality, which might lead to overlooking or not sufficiently penalizing disambiguation errors. Additionally, when compared with the sentences in the DIBIMT benchmark, the lower ambiguity in the two considered test sets could be a factor. In conclusion, whether it be due to limitations in the metrics or the datasets, our results suggest that the current common evaluation setting does not probe MT models' ability to translate ambiguous words effectively, highlighting the necessity for a benchmark specifically designed for it, such as DIBIMT.

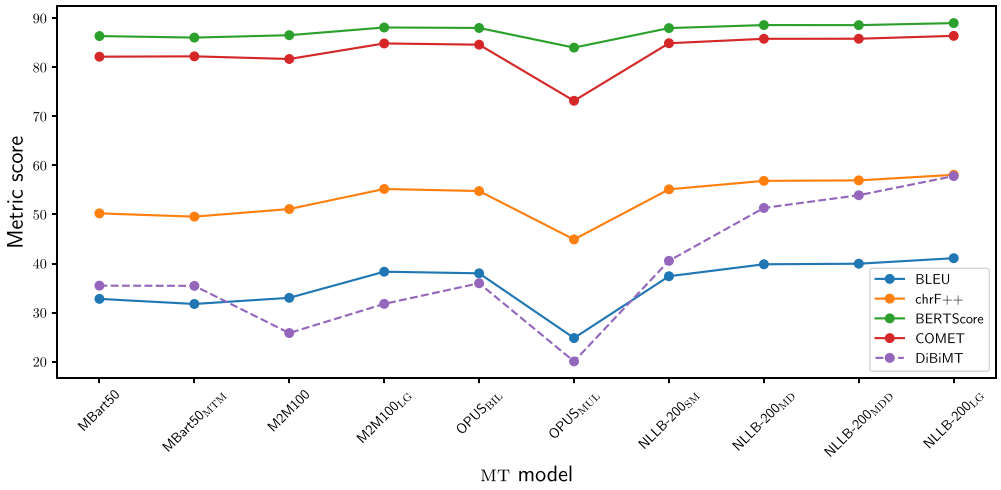
6.6 Findings of Our Experiments

Here we summarize the main findings of this section:

- The encoder of a Transformer-based encoder-decoder MT system provides a critical contribution to the overall disambiguation capabilities.



(a) Comparison between metrics scores (solid lines) on Flores-200 and the DiBiMT accuracy (dashed line).



(b) Comparison between metrics scores (solid lines) on Medline-2022 and the DiBiMT accuracy (dashed line).

Figure 7
Mean metrics scores for each model on the Flores-200 (7a) and Medline-2022 (7b) test sets, together with their DiBiMT accuracy. All scores have been averaged over all tested language pairs.

- In terms of disambiguation performance, MT systems' encoders learn representations that are less effective compared to BERT. However, this gap diminishes or even disappears with the use of larger MT models.
- Multilingual models seem to trade off their disambiguation capabilities in order to memorize more common senses in multiple languages.

- MT systems are intrinsically biased toward more frequent senses of ambiguous words, and increasing the beam size does not guarantee better disambiguation performance.
- MT evaluation needs an ad-hoc benchmark to assess how systems fare with the translation of ambiguous words, as the standard setting of evaluation is not fitting to be employed for investigating this phenomenon.

7. Recommendations for Future Improvements

In this section, we put forward some recommendations to the community for enhancing the MT capabilities based on the findings of our work:

- **Going beyond the surface and exploring the understanding capabilities of MT**: While MT systems have achieved tremendous progress in producing high-quality translations, this work highlights the crucial need to focus on their ability to deal with non-predominant meanings of ambiguous words. While some works already tackle this issue (Iyer, Chen, and Birch 2023; Iyer et al. 2023), there is still a long way to go to ensure that lexical ambiguity is properly addressed in MT.
- **Higher-resource languages still deserve attention:** While substantial efforts are being made to improve the availability of parallel corpora for low-resource languages, we observe that, despite its low number of `MISS` instances, a high-resource language such as Dutch exhibits the poorest performance in DiBiMT (see Table 5). In light of this counterintuitive finding, we encourage the MT community to assess and address the impact of lexical ambiguity on high-resource languages as well, because these, unexpectedly, may pose significant challenges.
- **Studying the interplay between LLMs and MT systems**: We observe a recent trend in the literature suggesting that adapting LLMs may yield superior results compared with training traditional MT architectures from scratch, while dropping the requirement of vast amounts of parallel corpora for training (Alves et al. 2024; Xu et al. 2024). In our work, we further explore this trend by assessing LLMs' ability to deal with lexical ambiguity. Interestingly, our experimental results show that Tower-7B, a language model for translation-related tasks built upon LLaMA 2, which we experimented with, outperforms LLaMA2-7B, LLaMA3-8B, and all open-source MT systems when evaluated against DiBiMT. In light of such findings, we suggest that the research community further investigates the potential of LLMs in MT.
- **Re-balancing the training data of MT systems**: One of the crucial obstacles that systems face when dealing with lexical ambiguity is sense infrequency. To translate words used with infrequent senses effectively, we encourage the community to explore artificially increasing the occurrences of such senses in training corpora. However, such data

modification could compromise the systems' ability to properly translate common senses. In such experimental scenarios, the DIBIMT benchmark can play a pivotal role in evaluating the effects of any adjustments to the training methodology and achieving an optimal balance.

8. Conclusion

In this article, we present DIBIMT, an entirely manually curated benchmark for investigating the ability of MT systems to deal with *lexical ambiguity*. DIBIMT covers eight language pairs, each composed of English and one of the following languages: Bulgarian, Chinese, Dutch, German, Italian, Russian, Slovene, and Spanish. We put forward a detailed study of the impact of lexical ambiguity on the automatic translation output by 22 systems, including both commercial and non-commercial MT systems, two commercial LLMs, and a set of open-access LLMs. We find that commercial systems consistently outperform their open-access counterparts, with GPT-4 achieving the best results overall. Among the open-access systems, Gemma2-9B stands out, achieving results that are substantially higher than all the rest and close to GPT-4. Moreover, although the performance of all systems significantly decreases when dealing with infrequent meanings and highly ambiguous words, we notice that top-performing models are more robust in handling the latter phenomenon. Conversely, all models struggle similarly with sense infrequency, which proves to be the most challenging obstacle in disambiguating ambiguous words. In this respect, we find that systems translate the focus word with the most frequent sense in about 50% of cases, while approximately 75% of the time they use a sense that is more frequent than the correct one. In light of these findings, and to counter the most prominent obstacle of sense infrequency, we put forward promising research directions for enhancing MT capabilities. Among our recommendations for future improvements, we include adapting LLMs to MT, since such systems do not necessarily require vast amounts of parallel corpora as traditional MT systems do. Furthermore, we suggest investigating and addressing the impact of lexical ambiguity on higher-resource languages, since these might pose unexpected challenges. We study the nature of errors produced by systems when translating ambiguous words by carrying out a manual error analysis, in which we identify four error patterns, namely, disambiguation errors, omissions, untranslated source words, and hallucinations. The first error category, disambiguation errors, is the most relevant to the DIBIMT benchmark, since this pattern consists of incorrect translations due to the erroneous choice of word meanings. Instead, omissions occur when a given model does not translate the focus word. We differentiate between severe and mild omissions depending on the impact of the omission on the overall quality and comprehensibility of the output translation. The third error pattern is that of untranslated source words, which are focus words reported as-is and therefore not translated into the target language. Finally, the last error pattern is represented by hallucinations, consisting of translations showing a significant semantic detachment from the source text. Differently from disambiguation errors, hallucinations are not possible senses of the focus word. Hallucinations also include a sub-pattern which we refer to as non-existing target words, that is, output words and expressions that do not exist in the vocabulary of the target language. Interestingly, based on our manual inspection, we find that, while disambiguation errors and hallucinations are commonly found in the output translations of all systems considered, untranslated source words and non-existing target words almost exclusively affect non-commercial systems.

In order to better investigate the ability of systems to deal with lexical ambiguity, we also carry out an extensive array of experiments. First, we investigate the impact of the encoder on the ability of a given MT model to disambiguate ambiguous words. Our results demonstrate that the encoder provides a vital contribution to the disambiguation capabilities of the corresponding architecture. Furthermore, we find that the representations learned by BERT are more effective than those learned by the encoders of MT systems when dealing with ambiguous words. Nonetheless, we show that capacity is key, and the encoders of larger MT systems are capable of surpassing BERT. We also investigate the presence of a trade-off between multilinguality and performance, a phenomenon which we refer to as *budgeting*, by evaluating a multilingual model and its bilingual counterparts against the DIBIMT benchmark. We find that the multilingual model OPUS_{MUL} consistently achieves worse disambiguation performance than its bilingual counterparts. We then examine the role of the decoding strategy in translating ambiguous words, focusing on the impact of varying beam sizes on a model's ability to disambiguate these words. Our experiments reveal that an increase in beam size does not necessarily improve performance, potentially because models are, in a high percentage of cases, not able to translate the focus word altogether. In this context, we show that MT systems exhibit an intrinsic bias toward more frequent senses of ambiguous words. Specifically, these systems might assign a higher probability to wrong translations of the ambiguous focus word despite the pool of decoded translations also including correct translations. Finally, we investigate the effectiveness of current MT evaluation metrics such as BLEU, chrF++, BERTScore, and COMET in assessing the ability of models to disambiguate ambiguous words, measuring the performance of MT systems on two popular test sets, namely, Flores-200 and Medline-2022. Crucially, we show that the standard evaluation setting is not fitting to be used for the investigation of disambiguation errors—whether it be due to the metrics not capturing them, or to the test sets not being particularly suited for the task—and highlight the need for an ad-hoc manually curated benchmark such as DIBIMT.

Our benchmark is available at `https://nlp.uniroma1.it/dibimt/`. In light of the major impact of lexical ambiguity on machine-generated translations, and despite the efforts made so far, we encourage the research community to devote particular care to improving the disambiguation capabilities of MT systems, thereby enabling a significantly better translation performance.

Appendix A. Annotation Guidelines

This work aims to create a novel entirely manually-curated evaluation benchmark called DiBiMT which allows semantic biases in MT to be investigated.

With this aim in view, you receive a spreadsheet that contains approximately 1,000 automatically-extracted instances, each comprising the following data: i) a lemma and its part of speech (PoS), associated with a definition derived from either WordNet or Wiktionary; ii) a sentence in English containing a focus word for which some good and bad translation candidates derived from BabelNet are provided.

For each instance, good translation candidates are located on the same line as the definition and the example, whereas bad translation candidates can be found on the line below. From a translation perspective, a good candidate can be described as a correct translation into the target language for the English focus word. Instead, a bad candidate is an incorrect translation for the English focus word in the given context.

Annotators are asked to verify: i) the correctness of the good translation candidates and add new good translations if deemed necessary; ii) the incorrectness of the bad

translation candidates provided. Furthermore, annotators are required to adopt the following guidelines. Do not annotate idioms and mark them with the tag *IDIOM*. Do not annotate instances in which the semantic context does not allow us to unequivocally determine the meaning of the focus word and label these with the tag *X*. Discard instances containing proper names as focus words (e.g., *The military campaign near that creek was known as "The battle of Bull *Run*"*). Mark with the tag *DISCUSS* challenging instances which should be discussed during joint sessions. Annotators are allowed to include cross-PoS candidates, that is, candidates whose PoS is different from that of the focus word and, when this is the case, annotators are required to include the candidate in square brackets in the following way: [candidate_with_different_pos | P]. Annotators are asked to do the same for multi-word expressions as well: [multi_word_expression | P]. Annotators can specify the PoS by adding a letter (either, *n*, *a*, *v* or *r*) after | P.

Once the aforementioned annotation steps are finalized, annotators are required to study and classify as either `GOOD` or `BAD` all so-called `MISS` cases, that is, translation candidates proposed by the MT models and LLMs considered and classified as neither `GOOD` nor `BAD`. Please note that we expect that in only a small percentage of `MISS` cases is the target translation actually missing, such as in omissions (see Section 5).

Appendix B. Discussion of Index-based Weighting in SFI

As illustrated in Section 3.4, SFI weighs instances based on their ambiguous word meaning's frequency index, that is, $\mu_{\lambda_p}(\sigma)$. This metric could be reformulated to use sense frequencies instead of their indices to weigh the item's contribution to the metric score. In this section, we discuss our reasoning for using an index-based weighting instead of a frequency-based one.

Our assumption is that computing frequencies for very low-frequency meanings is inherently bound to produce a very noisy and unreliable estimation, especially when considering that there are very few manually annotated corpora for WSD.[34] Therefore, given that word meanings tend to follow a Zipfian distribution, and that our very infrequent senses are likely to have wrong estimations, we decide to leverage Zipf's law and use the indices instead of the raw frequency, as it should display better stability across different corpora compared to raw frequency counts, providing a much more robust and reliable score.

Appendix C. DiBiMT Experimental Setup

In this section, we report the generation parameters used when translating the sentences of DiBiMT with the open-source MT systems; the API call we use for translating with DeepL, Google, GPT-3.5$_{\text{TURBO}}$, and GPT-4; and the prompt provided to GPT systems and LLMs.

Generation Parameters. For open-source models, we use a beam size of 5, with early stopping. We apply no length penalty and return the sequence with the lowest perplexity, among those generated using the beam search algorithm.

34 WordNet computes its sense orderings via SemCor, arguably the most used manually annotated English WSD dataset, which is composed of sentences coming from news articles of the 1960s.

Table 18
Breakdown of UNK instances.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	1.80	2.85	1.50	1.05	1.65	1.20	0.75	0.60	1.42
DeepL	1.95	2.40	1.20	1.20	2.25	1.35	0.90	0.45	1.46
MBart50	1.80	1.80	1.05	0.75	1.05	–	0.60	0.30	1.05
MBart50$_{MTM}$	1.35	1.20	1.05	0.45	0.60	–	0.45	0.75	0.84
M2M100	0.75	1.50	1.20	0.60	0.45	0.60	0.45	0.90	0.81
M2M100$_{LG}$	1.20	1.50	0.90	1.05	0.75	0.90	0.75	0.60	0.96
OPUS$_{BIL}$	1.80	2.10	0.90	1.35	2.10	1.20	0.75	–	1.46
OPUS$_{MUL}$	0.75	1.20	1.05	0.30	0.60	0.45	0.45	0.30	0.64
NLLB-200$_{SM}$	1.50	1.95	0.60	0.60	0.45	0.45	1.05	0.90	0.94
NLLB-200$_{MD}$	1.35	2.10	0.60	0.60	1.05	0.45	0.90	0.90	0.99
NLLB-200$_{MDD}$	1.05	1.80	1.20	0.90	0.60	0.75	0.90	0.90	1.01
NLLB-200$_{LG}$	1.20	1.95	1.35	0.75	0.90	0.60	0.75	0.45	0.99
Llama2-7b	1.50	1.80	1.20	1.20	0.30	0.45	0.45	0.30	0.90
Llama3-8B	1.50	1.80	1.35	1.05	0.90	0.45	1.05	0.60	1.09
Mistral-7b	2.40	2.40	1.80	1.35	2.25	1.20	2.25	0.75	1.80
Gemma-2B	0.30	1.80	0.75	0.90	1.20	0.45	0.00	0.15	0.69
Gemma-7B	0.90	2.25	2.70	0.45	2.10	1.20	2.25	0.90	1.59
Gemma2-9B	0.90	2.10	1.65	1.20	2.25	0.75	1.20	0.90	1.37
Phi3-mini	1.65	2.25	1.35	0.15	0.90	0.15	0.45	0.00	0.86
Tower-7B	0.90	3.45	1.95	1.35	2.25	0.45	1.20	0.00	1.44
GPT-3.5$_{TURBO}$	0.75	2.70	2.40	1.20	2.70	0.75	1.20	0.75	1.56
GPT-4	1.50	2.85	2.25	1.20	3.15	0.75	0.90	0.75	1.67
Mean	1.31	2.08	1.36	0.89	1.38	0.73	0.89	0.58	1.15

API Calls. We generate translations into all our target languages with GPT-3.5$_{TURBO}$ and GPT-4 using the API made available by OpenAI.[35] For both models, we use the following instruction-based prompt: "Translate the following English text to {lang}: {source_text}", where {lang} and {source_text} are two variables containing a given target language and the source sentence to be translated, respectively. In both cases, we adopt the default hyperparameters.

Appendix D. DiBiMT Additional Results

In this section, we report additional information regarding the overall results of systems on the DiBiMT benchmark, discussed in Section 4.

Appendix D.1 UNK Instances

A breakdown of UNK instances is reported in Table 18.

35 https://platform.openai.com/.

Appendix D.2 MFS and MFS+

Table 19 reports the scores of MFS and MFS+ metrics, indicating the percentage of times that a system translates ambiguous words with their most frequent sense (MFS), or with senses more frequent than the correct one (MFS+). As can be seen from the *Mean* column, most systems obtain roughly the same scores, irrespective of their performance; indeed, even GPT-4, which is the best among the tested systems, reports MFS and MFS+ scores in line with the others. This suggests that almost all systems[36] tend to default to more frequent senses at the same rate.[37]

Appendix D.3 Noun vs. Verbs

Tables 20 and 21 report the accuracy scores of all systems when measured only on nouns and verbs, respectively.

Appendix E. WSD Systems Training – Experimental Setup

In this section, we describe the experimental setup used for training WSD systems.

Appendix E.1 Data

We use SemCor and WNG for training and SemEval-2007 for development:

- **SemCor** corpus is typically used for training WSD systems, which contains 33,362 sentences, totaling 226,036 instances, which have been manually annotated with their WordNet sense.
- **WNG** contains 117,659 definitions and 48,318 examples coming from WordNet, manually annotated with their sense.
- **SemEval-2007** comprises 455 sense-annotated instances sourced from articles in the Wall Street Journal Corpus (Paul and Baker 1992). Unlike other datasets, the annotated instances in SemEval-2007 are exclusively nouns or verbs.

Appendix E.2 Model Architecture

Our architecture is based on that of Amuse-WSD (Orlando et al. 2021), with some minor modifications. Specifically, given a word in context w, we first encode w and extract the hidden state of the last layer of the encoder; then, we apply batch normalization to

36 We note that M2M100 systems prefer the most frequent sense at a lower rate, compared with other systems.

37 We clarify that MFS and MFS+ are computed considering only the `BAD` instances, and therefore the scores of different systems are not computed on the same number of instances.

Table 19
Frequency analysis. MFS represents the percentage of times the model mistakenly translates the focus word into a lexicalization belonging to the Most Frequent Sense associated with λ_p. MFS+, instead, is the percentage of times the wrong translation belongs to any synset that is more frequent than the focus one.

Model	Metric	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	MFS	58.37	60.50	57.48	41.76	48.88	42.86	51.75	47.66	51.16
Google	MFS+	80.54	75.21	76.38	72.53	69.96	75.89	76.57	76.56	75.45
DeepL	MFS	52.73	57.22	60.23	42.86	46.34	49.35	50.82	51.35	51.36
DeepL	MFS+	72.73	74.87	80.11	75.32	71.14	75.32	77.87	77.48	75.61
MBart50	MFS	53.75	48.85	57.19	42.37	49.24	–	56.09	51.35	51.26
MBart50	MFS+	80.31	65.23	81.75	80.53	77.48	–	78.53	81.08	77.84
MBart50$_{\text{MTM}}$	MFS	53.80	57.00	61.73	43.08	50.00	–	52.56	57.94	53.73
MBart50$_{\text{MTM}}$	MFS+	81.19	76.45	84.12	82.21	79.92	–	77.56	85.98	81.06
M2M100	MFS	56.63	59.81	53.33	43.80	51.59	44.80	52.75	49.66	51.55
M2M100	MFS+	80.65	76.32	80.33	80.23	81.98	76.80	79.61	79.19	79.39
M2M100$_{\text{LG}}$	MFS	51.72	59.94	56.58	39.35	53.92	44.88	51.95	45.96	50.54
M2M100$_{\text{LG}}$	MFS+	78.68	76.09	82.89	77.98	83.66	76.38	79.88	78.26	79.23
OPUS$_{\text{BIL}}$	MFS	51.80	57.26	57.66	41.22	48.81	49.57	53.82	–	51.45
OPUS$_{\text{BIL}}$	MFS+	82.04	73.74	83.48	75.95	76.11	79.13	78.59	–	78.43
OPUS$_{\text{MUL}}$	MFS	56.49	68.17	57.88	50.76	57.68	54.74	55.02	54.17	56.86
OPUS$_{\text{MUL}}$	MFS+	85.50	84.08	82.19	85.28	81.65	81.05	77.70	79.17	82.08
NLLB-200$_{\text{SM}}$	MFS	56.73	59.15	57.61	42.21	51.54	46.15	50.35	53.54	52.16
NLLB-200$_{\text{SM}}$	MFS+	81.82	76.47	81.52	79.90	77.09	76.92	77.11	82.83	79.21
NLLB-200$_{\text{MD}}$	MFS	55.17	58.10	55.16	44.91	50.25	47.19	49.13	50.67	51.32
NLLB-200$_{\text{MD}}$	MFS+	81.77	75.49	76.23	75.45	74.11	71.91	77.39	77.33	76.21
NLLB-200$_{\text{MDD}}$	MFS	57.81	57.31	50.97	40.91	50.79	50.62	51.74	52.00	51.52
NLLB-200$_{\text{MDD}}$	MFS+	81.77	74.70	76.21	73.30	73.82	76.54	80.87	78.67	76.98
NLLB-200$_{\text{LG}}$	MFS	54.82	57.38	54.35	44.00	50.50	52.70	43.26	63.49	52.56
NLLB-200$_{\text{LG}}$	MFS+	81.93	72.57	75.00	73.33	71.29	79.73	78.14	87.30	77.41
Llama2-7b	MFS	55.56	62.27	52.88	48.82	46.58	37.50	52.53	54.10	51.28
Llama2-7b	MFS+	80.70	78.18	83.25	83.46	70.55	75.00	78.28	77.05	78.31
Llama3-8B	MFS	59.09	59.90	56.35	35.71	46.31	41.18	52.36	64.62	51.94
Llama3-8B	MFS+	83.77	76.33	79.70	73.81	67.79	75.00	77.83	86.15	77.55
Mistral-7b	MFS	52.56	57.08	50.53	45.91	42.78	32.88	46.67	45.83	46.78
Mistral-7b	MFS+	82.69	76.71	77.89	81.13	68.56	65.75	73.33	59.72	73.22
Gemma-2B	MFS	57.32	58.20	54.17	38.10	46.43	42.22	45.74	47.22	48.67
Gemma-2B	MFS+	78.05	71.96	80.36	68.57	67.26	77.78	69.15	63.89	72.13
Gemma-7B	MFS	56.74	56.46	55.87	40.00	50.27	30.12	45.41	41.25	47.02
Gemma-7B	MFS+	80.34	75.12	84.36	75.45	71.35	43.37	71.50	57.50	69.87
Gemma2-9B	MFS	56.64	63.58	61.83	30.56	44.03	54.39	46.94	57.14	51.89
Gemma2-9B	MFS+	76.11	77.16	83.21	68.06	60.45	80.70	74.15	84.13	75.50
Phi3-mini	MFS	51.30	61.84	55.98	44.09	42.08	41.46	60.25	59.38	52.05
Phi3-mini	MFS+	77.27	78.26	79.43	77.17	69.40	70.73	82.61	71.88	75.84
Tower-7B	MFS	61.07	59.52	52.47	37.93	44.38	34.78	51.46	57.78	49.92
Tower-7B	MFS+	80.92	73.81	77.78	70.69	63.31	73.91	74.27	73.33	73.50
GPT-3.5$_{\text{TURBO}}$	MFS	54.48	63.58	54.60	40.68	43.14	45.07	52.49	50.75	50.60
GPT-3.5$_{\text{TURBO}}$	MFS+	74.63	73.99	77.91	70.34	64.05	74.65	76.24	82.09	74.24
GPT-4	MFS	55.65	64.29	54.84	38.54	42.75	44.93	51.43	55.88	51.04
GPT-4	MFS+	75.00	75.32	77.42	66.67	61.07	72.46	72.00	80.88	72.60
Mean	MFS	55.47	59.43	55.89	41.71	48.10	44.37	51.11	52.94	51.21
Mean	MFS+	79.93	75.37	80.07	75.79	71.91	73.95	76.78	77.17	76.44

Table 20
Noun accuracy.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	61.13	58.23	54.65	64.74	50.52	45.80	45.62	51.66	54.04
DeepL	81.08	66.77	69.60	69.52	54.45	55.42	56.72	55.71	63.66
MBart50	36.52	36.17	38.76	38.28	34.69	–	25.97	33.33	34.82
MBart50$_{\text{MTM}}$	36.53	39.38	34.90	37.96	34.57	–	25.20	35.71	34.89
M2M100	28.38	32.67	25.51	29.74	16.50	19.81	21.74	28.37	25.34
M2M100$_{\text{LG}}$	32.47	34.69	32.23	35.38	24.59	24.89	25.27	30.89	30.05
OPUS$_{\text{BIL}}$	37.81	41.40	36.07	40.83	33.46	32.76	35.19	–	36.79
OPUS$_{\text{MUL}}$	26.13	29.57	23.81	26.04	11.63	18.01	19.11	25.00	22.41
NLLB-200$_{\text{SM}}$	38.27	45.58	40.29	49.36	34.76	31.77	33.47	43.81	39.66
NLLB-200$_{\text{MD}}$	53.53	52.78	52.94	55.31	42.78	43.16	44.02	55.08	49.95
NLLB-200$_{\text{MDD}}$	55.78	54.11	54.35	58.20	45.63	43.22	48.02	57.00	52.04
NLLB-200$_{\text{LG}}$	62.95	59.53	62.73	62.20	44.95	50.00	52.00	59.90	56.78
Llama2-7b	55.56	54.07	53.15	57.05	45.64	35.88	40.87	46.97	48.65
Llama3-8B	66.26	59.04	61.54	69.64	53.97	43.51	55.27	52.17	57.68
Mistral-7b	61.90	52.02	56.83	57.60	45.69	38.75	40.09	43.79	49.58
Gemma-2B	44.33	49.77	39.77	52.99	42.35	26.92	27.72	38.96	40.35
Gemma-7B	50.47	54.88	52.29	58.02	48.57	29.79	32.41	36.67	45.39
Gemma2-9B	77.29	68.38	72.00	81.09	67.36	56.80	64.96	62.50	68.80
Phi3-mini	61.19	55.14	51.29	51.08	38.98	25.32	31.71	32.76	43.43
Tower-7B	69.05	65.43	65.58	67.81	56.71	45.71	61.66	49.11	60.13
GPT-3.5$_{\text{TURBO}}$	72.79	67.12	68.04	71.76	65.52	52.50	65.41	64.61	65.97
GPT-4	75.17	68.24	70.97	78.44	68.20	59.24	65.52	65.93	68.96
Mean	53.85	52.04	50.79	55.14	43.71	38.96	41.72	46.19	47.80

obtain $\boldsymbol{e}_w \in \mathbb{R}^d$. Finally, we use a two-layer fully connected neural network to assign the correct sense to the word in context. More formally:

$$\boldsymbol{e}_w = \text{BatchNorm}\left(\boldsymbol{l}_w^{-1}\right)$$
$$\boldsymbol{h}_w = \text{SiLU}(\boldsymbol{W}_h \boldsymbol{e}_w)$$
$$\boldsymbol{o}_w = \boldsymbol{W}_o \boldsymbol{h}_w$$

where $\boldsymbol{l}_w^{-1}$ is the last hidden state of the transformer, BatchNorm$(\cdot)$ is the batch normalization operation, and $\text{SiLU}(x) = x \cdot \text{sigmoid}(x)$ is the Sigmoid-weighted Linear Unit (Elfwing, Uchibe, and Doya 2017). $\boldsymbol{W}_h$ and $\boldsymbol{W}_o$ are the weights of the first and second layers of the fully connected, and the bias is 0. In all experiments, the weights of the encoders are frozen, and the only trainable parameters are those in $\boldsymbol{W}_h$ and $\boldsymbol{W}_o$.

Appendix E.3 Training Objective

Sense boundaries are not always clearly defined. Indeed, in the training datasets utilized, there are some cases in which annotators have deemed multiple senses appropriate for the same instance. For this reason, Conia and Navigli (2021) frame WSD as a

Table 21
Verb accuracy.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	63.90	56.59	61.87	72.56	64.32	57.01	43.43	61.71	60.17
DeepL	82.88	62.50	71.89	78.55	56.70	71.23	46.15	71.76	67.71
MBart50	31.25	25.88	37.91	44.39	45.45	–	19.33	50.35	36.37
MBart50$_{\text{MTM}}$	30.89	28.65	42.57	42.94	44.79	–	18.99	42.95	35.97
M2M100	23.03	21.21	28.24	35.14	25.66	26.76	13.28	38.36	26.46
M2M100$_{\text{LG}}$	30.96	30.33	37.37	42.93	31.11	35.33	21.82	43.09	34.12
OPUS$_{\text{BIL}}$	31.33	32.57	38.60	41.83	35.75	40.00	24.60	–	34.95
OPUS$_{\text{MUL}}$	13.53	19.88	18.31	20.43	7.20	10.19	17.14	24.75	16.43
NLLB-200$_{\text{SM}}$	37.75	38.08	47.06	52.87	44.17	44.53	22.73	45.99	41.65
NLLB-200$_{\text{MD}}$	54.59	50.82	58.26	63.54	47.59	53.64	36.94	57.83	52.90
NLLB-200$_{\text{MDD}}$	61.40	49.18	65.07	60.64	52.98	60.25	39.26	59.78	56.07
NLLB-200$_{\text{LG}}$	64.65	51.20	65.13	72.16	50.60	59.51	44.77	64.37	59.05
Llama2-7b	38.17	43.59	49.72	48.31	45.38	30.67	32.43	49.33	42.20
Llama3-8B	61.46	53.00	53.02	65.27	57.82	35.58	33.75	59.62	52.44
Mistral-7b	53.33	46.88	52.74	57.23	43.14	32.80	36.36	51.43	46.74
Gemma-2B	47.66	41.96	40.19	48.15	44.88	28.57	26.67	48.89	40.87
Gemma-7B	51.01	46.03	52.47	61.11	54.44	38.75	39.00	53.25	49.51
Gemma2-9B	71.50	62.26	73.73	86.07	70.59	74.82	54.23	72.84	70.75
Phi3-mini	60.80	48.76	50.26	45.87	41.09	19.35	23.44	38.71	41.03
Tower-7B	71.94	64.09	68.20	77.72	59.77	40.32	48.97	55.74	60.84
GPT-3.5$_{\text{TURBO}}$	74.90	65.68	70.80	79.63	68.50	63.70	54.80	72.45	68.81
GPT-4	79.68	72.27	72.65	82.35	75.13	70.11	58.10	78.60	73.61
Mean	51.66	45.97	52.55	58.17	48.50	44.66	34.37	54.37	48.78

multi-label classification problem, where a model is trained to maximize the probability of *all* the appropriate senses of a word in context. Furthermore, they find it beneficial to integrate relational information into the training algorithm. In this respect, they extend the set of appropriate senses for a word in context by exploiting the semantic connections between pairs of senses. More formally, given a focus word w, let S_w be the set of candidate senses of w, and $\hat{S}_w \subseteq S_w$ be the set of correct senses. Let R be the set of semantic connections (e.g., hypernymy or hyponymy, among others, between any two senses). We can now define $\hat{S}_w^+ = \hat{S}_w \cup \{s_j : (s_i, s_j) \in R, s_i \in \hat{S}_w\}$ as the new set of appropriate senses for w, obtained by extending $\hat{S}_w$ to include every sense s_j that is connected to any sense $s_i \in \hat{S}_w$ by means of a semantic connection in R.

Our WSD systems are thus trained to minimize the binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}}(w, \hat{S}_w^+) = -\sum_{s \in \hat{S}_w^+} \log(y_s) \tag{E.3}$$

$$-\sum_{s \in S_w^+ \setminus \hat{S}_w^+} \log(1 - y_s) \tag{E.4}$$

Table 22
Accuracy values for the comparison in Figure 5.

Model	WSD	DiBiMT
MBart50	33.07	35.49
$\text{MBart50}_{\text{MTM}}$	34.76	35.45
M2M100	27.35	25.83
$\text{M2M100}_{\text{LG}}$	33.29	31.78
OPUS_{BIL}	25.93	35.98
OPUS_{MUL}	10.04	20.06
$\text{NLLB-200}_{\text{SM}}$	39.06	40.55
$\text{NLLB-200}_{\text{MD}}$	46.96	51.83
$\text{NLLB-200}_{\text{MDD}}$	49.95	53.92
$\text{NLLB-200}_{\text{LG}}$	55.63	57.82

where $S_w^+ = S_w \cup \{s_j : (s_i, s_j) \in R, s_i \in S_w\}$ and y_s is the probability assigned to sense s by the WSD system.

Appendix F. Additional Experimental Results

In this section, we report additional information regarding the experiments of Section 6. In Table 22 there are the exact numerical values of the comparison shown in Figure 5. Figures 8–19 show the comparison between metrics scores on Flores-200 and Medline-2022, and the DiBiMT accuracy. Differently from Figure 7, the scores are not averaged across languages, and therefore we have a different figure per each language direction.

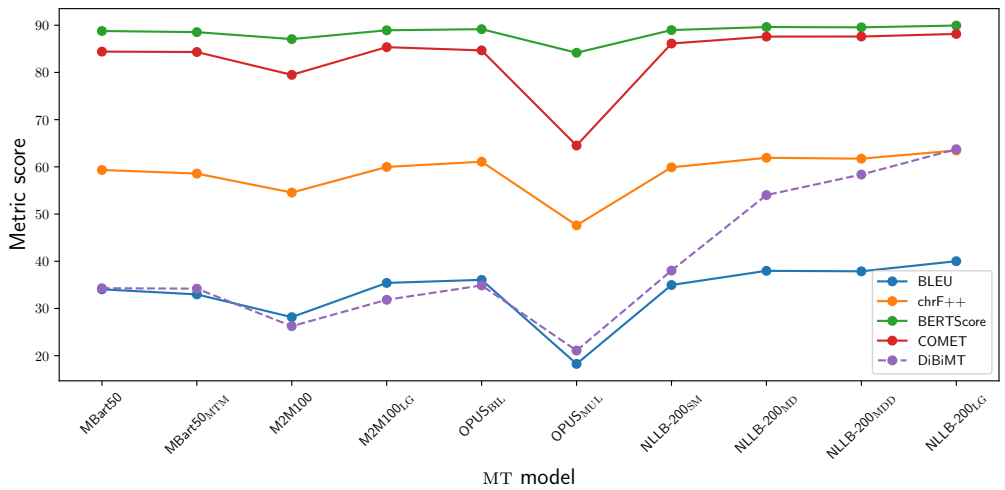


Figure 8
Metrics scores for each model on the Flores-200 test set, together with their DiBiMT accuracy, when translating from English into German.

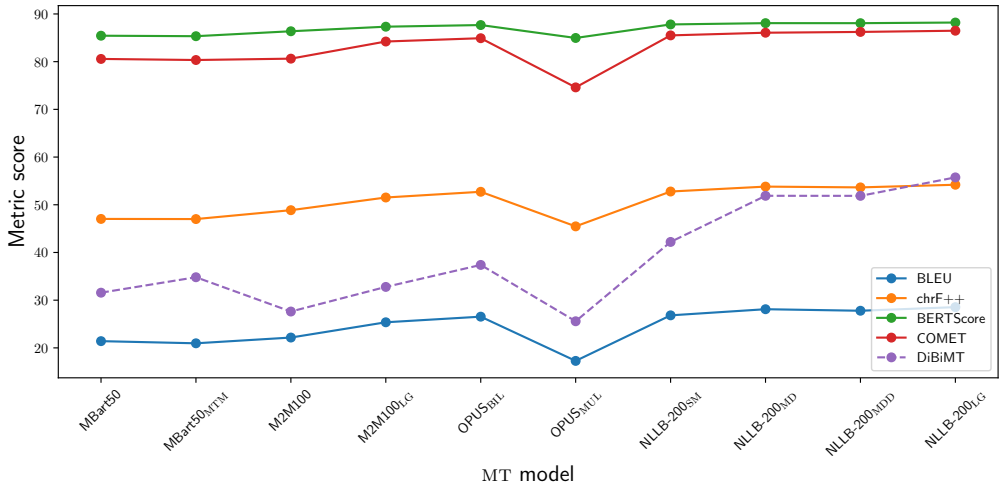


Figure 9
Metrics scores for each model on the Flores-200 test set, together with their DiBiMT accuracy, when translating from English into Spanish.

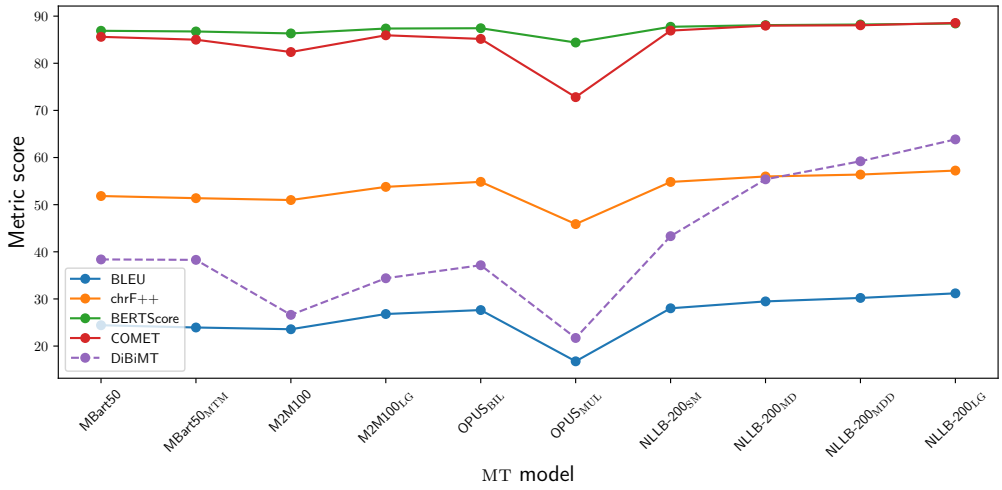


Figure 10
Metrics scores for each model on the Flores-200 test set, together with their DiBiMT accuracy, when translating from English into Italian.

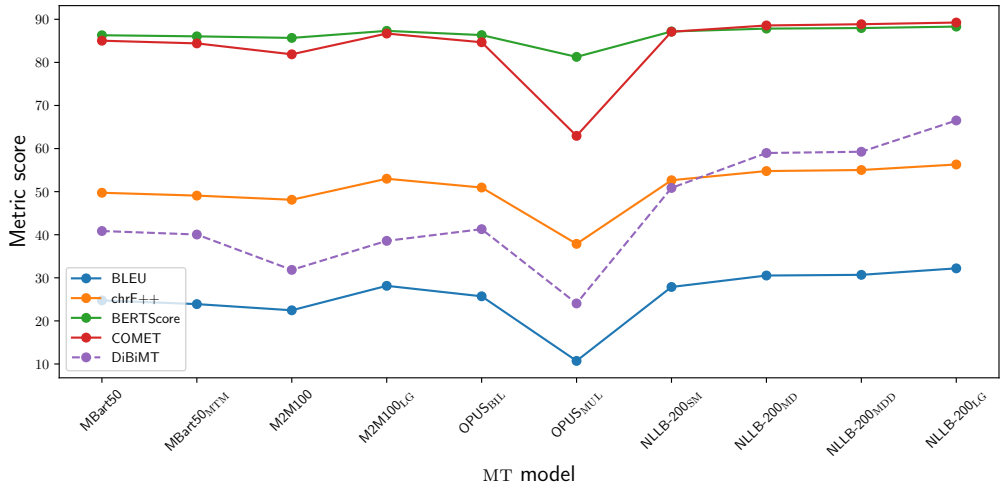


Figure 11
Metrics scores for each model on the Flores-200 test set, together with their DIBIMT accuracy, when translating from English into Russian.

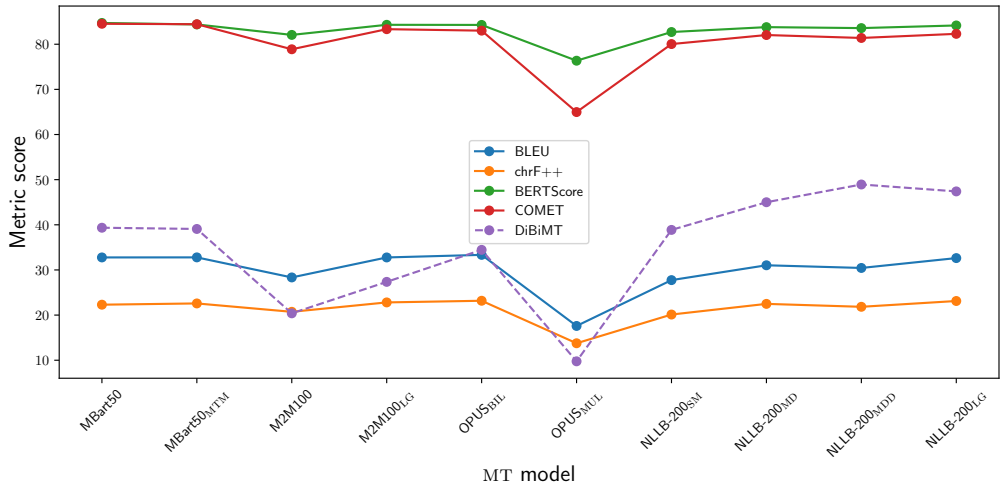


Figure 12
Metrics scores for each model on the Flores-200 test set, together with their DIBIMT accuracy, when translating from English into Chinese.

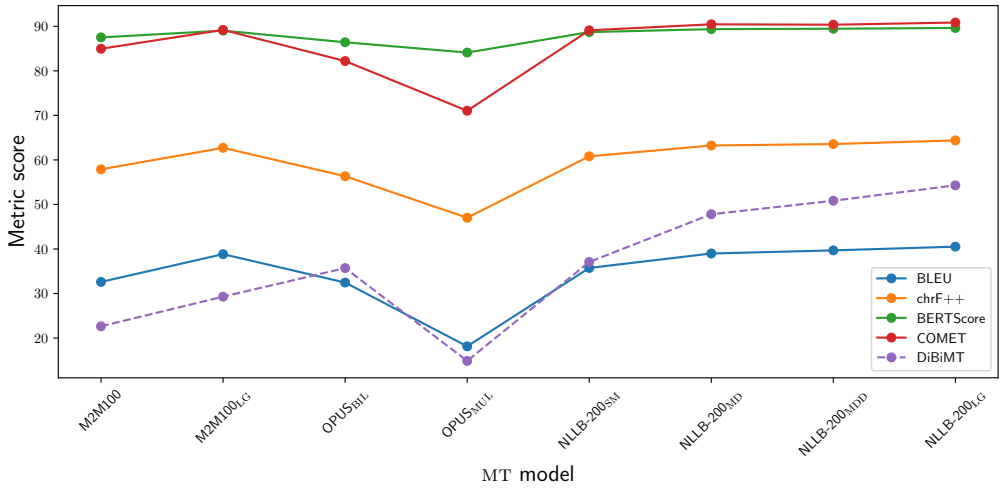


Figure 13
Metrics scores for each model on the Flores-200 test set, together with their DiBiMT accuracy, when translating from English into Bulgarian.

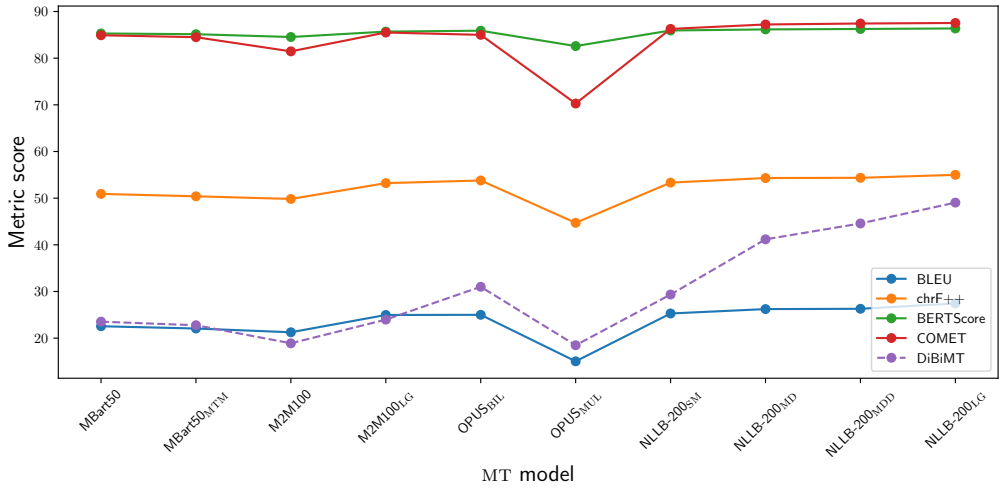


Figure 14
Metrics scores for each model on the Flores-200 test set, together with their DiBiMT accuracy, when translating from English into Dutch.

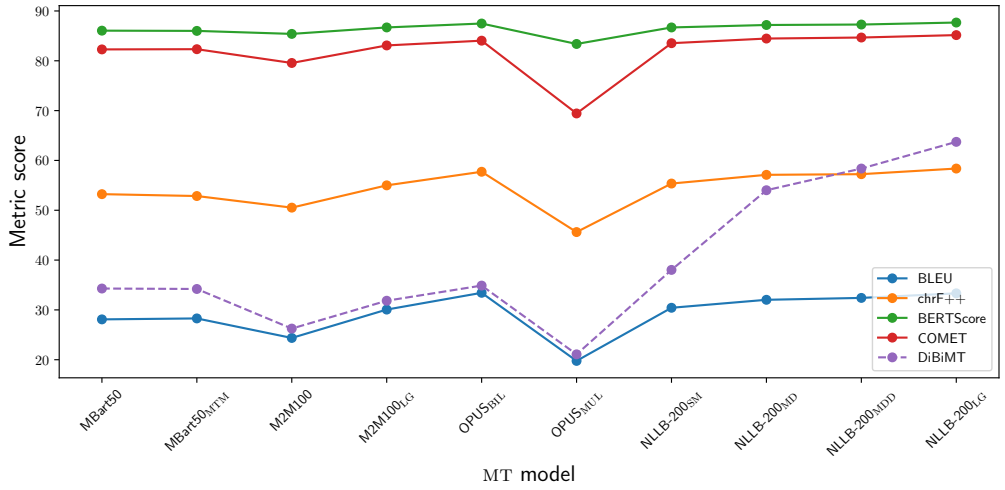


Figure 15
Metrics scores for each model on the Medline-2022 test set, together with their DiBiMT score, when translating from English into German.

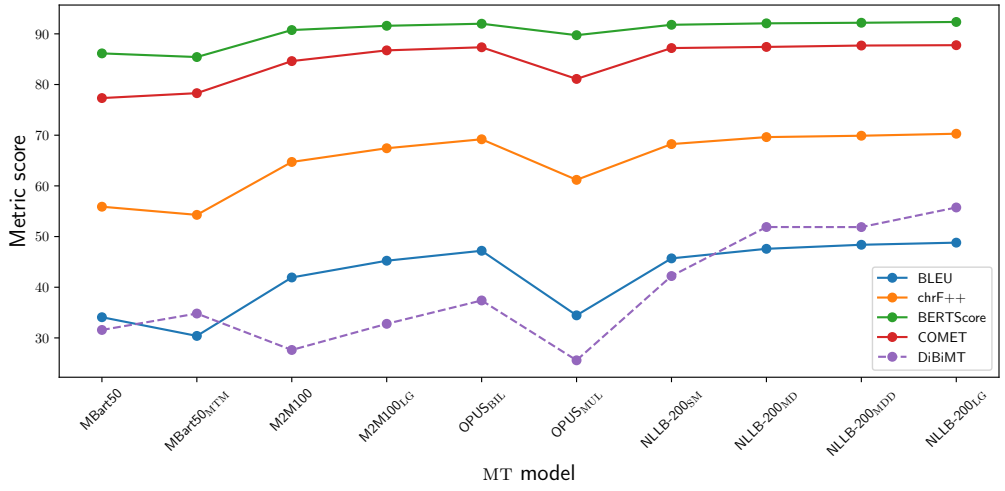


Figure 16
Metrics scores for each model on the Medline-2022 test set, together with their DiBiMT score, when translating from English into Spanish.

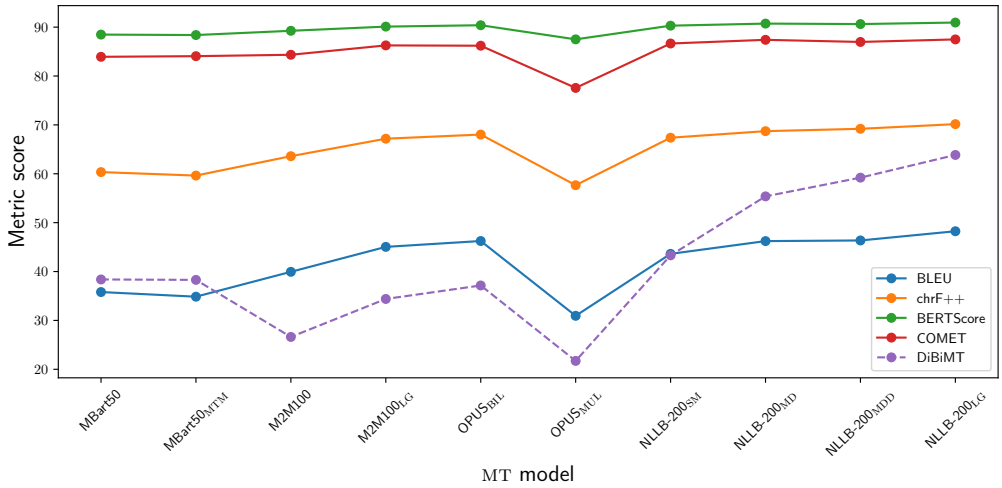


Figure 17
Metrics scores for each model on the Medline-2022 test set, together with their DIBIMT score, when translating from English into Italian.

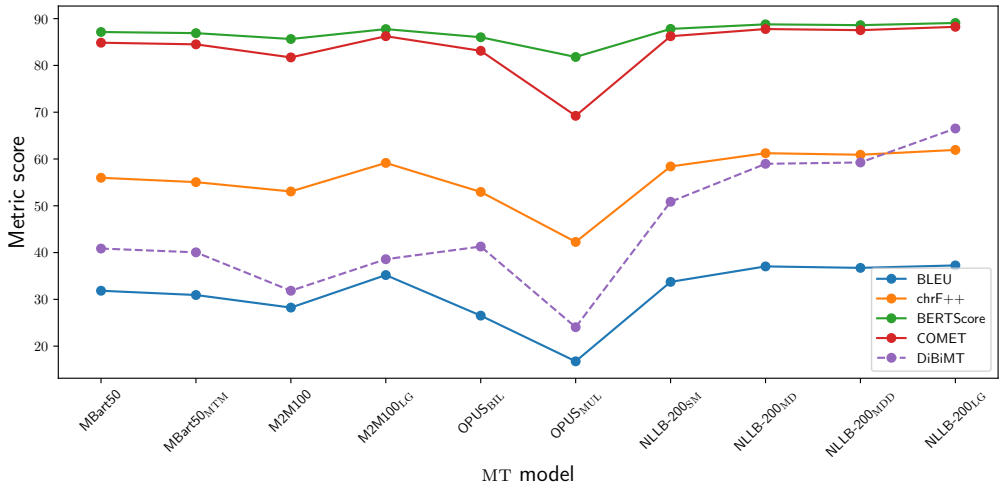


Figure 18
Metrics scores for each model on the Medline-2022 test set, together with their DIBIMT score, when translating from English into Russian.

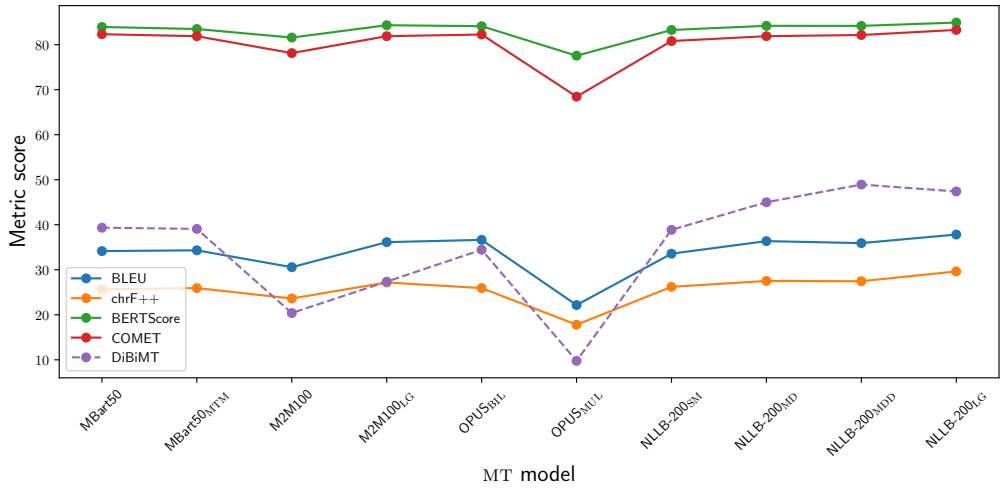


Figure 19
Metrics scores for each model on the Medline-2022 test set, together with their DIBIMT score, when translating from English into Chinese.

Appendix G. Additional Tables

Table 23
MISS % after the manual refinement process described in Section 4.3.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	9.01	11.42	7.00	12.42	21.65	27.77	21.75	13.88	15.61
DeepL	4.43	14.29	7.44	10.47	15.18	29.94	22.39	18.37	15.31
MBart50	25.19	22.14	28.94	33.08	34.55	–	38.46	49.47	33.12
MBart50$_{\text{MTM}}$	29.79	31.56	30.76	36.45	34.39	–	39.16	50.00	36.02
M2M100	42.45	31.66	36.72	42.68	46.08	47.36	42.62	45.39	41.87
M2M100$_{\text{LG}}$	28.98	26.64	28.74	31.67	35.95	40.09	33.84	32.58	32.31
OPUS$_{\text{BIL}}$	21.22	11.94	19.36	31.91	31.55	40.52	28.40	–	26.41
OPUS$_{\text{MUL}}$	49.85	40.67	43.48	60.60	55.20	59.49	50.30	58.35	52.24
NLLB-200$_{\text{SM}}$	31.96	18.50	25.49	38.61	43.83	50.45	39.09	49.92	37.23
NLLB-200$_{\text{MD}}$	31.91	18.53	24.28	38.61	45.45	48.64	40.85	46.60	36.86
NLLB-200$_{\text{MDD}}$	29.39	18.17	23.37	34.64	43.59	45.62	37.22	41.60	34.20
NLLB-200$_{\text{LG}}$	29.29	16.06	22.64	32.33	41.91	45.55	36.25	43.37	33.42
Llama2-7b	48.55	32.67	39.45	58.42	59.70	68.98	51.96	68.87	53.58
Llama3-8B	33.33	25.50	27.81	40.76	49.17	61.14	39.85	60.03	42.20
Mistral-7b	42.40	32.41	34.66	42.86	46.32	56.75	47.39	58.61	45.17
Gemma-2B	54.74	45.34	57.25	67.47	54.93	81.93	61.17	81.68	63.06
Gemma-7B	45.08	33.28	41.45	59.34	41.96	66.46	51.53	65.66	50.59
Gemma2-9B	31.77	25.88	25.00	33.38	34.20	53.47	42.94	46.44	36.64
Phi3-mini	39.79	31.90	35.41	62.76	53.71	83.48	65.66	86.66	57.42
Tower-7B	32.22	24.07	24.62	35.26	37.88	74.85	39.61	74.06	42.82
GPT-3.5$_{\text{TURBO}}$	21.15	18.18	16.90	27.47	28.97	47.73	28.83	33.69	27.86
GPT-4	15.83	17.59	14.88	25.64	29.72	41.84	29.05	26.74	25.16
Mean	31.74	24.93	27.98	38.95	40.27	53.60	40.38	50.09	38.49

Table 24
MISS % before the manual refinement process described in Section 4.3.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	39.27	29.48	28.70	52.80	42.03	81.83	39.67	81.53	49.41
DeepL	45.92	35.61	36.30	56.04	40.49	85.11	43.29	84.08	53.36
MBart50	40.06	43.20	41.87	53.54	48.80	–	46.84	84.36	51.24
MBart50$_{MTM}$	42.56	43.46	43.93	54.59	48.87	–	49.25	84.36	52.43
M2M100	50.68	44.02	44.14	55.34	52.26	82.16	50.98	80.18	57.47
M2M100$_{LG}$	43.35	38.01	38.83	49.77	45.63	81.23	43.59	77.78	52.27
OPUS$_{BIL}$	37.10	26.36	30.62	51.21	43.47	81.20	39.61	–	44.22
OPUS$_{MUL}$	54.05	46.46	45.93	65.26	57.36	84.41	55.26	81.53	61.28
NLLB-200$_{SM}$	43.07	32.63	38.20	57.23	54.52	82.88	47.44	84.08	55.01
NLLB-200$_{LG}$	48.42	34.75	40.33	60.84	57.40	84.71	51.13	86.34	57.99
NLLB-200$_{MD}$	48.42	33.74	39.97	57.92	55.87	84.98	48.64	85.44	56.87
NLLB-200$_{LG}$	50.83	32.27	41.20	60.84	55.05	85.89	48.57	86.79	57.68
LLaMA2-7B	57.53	42.51	50.98	70.50	68.32	88.46	60.75	88.74	65.97
LLaMA3-8B	52.72	38.37	42.36	62.58	61.99	87.26	50.76	88.16	60.53
Mistral-7B	55.35	44.71	48.18	60.15	56.36	86.92	57.90	86.77	62.04
Gemma-2B	65.06	54.63	64.06	75.98	62.59	92.04	67.07	93.10	71.82
Gemma-7B	58.35	45.80	51.82	73.30	53.51	84.68	59.12	84.71	63.91
Gemma2-9B	55.64	39.40	44.73	65.41	53.73	87.67	58.16	86.94	61.46
Phi3-mini	56.13	42.88	47.36	71.88	62.05	93.25	68.87	95.05	67.18
Tower-7B	53.70	40.92	44.11	61.82	51.28	90.70	55.29	91.60	61.18
GPT-3.5$_{TURBO}$	47.44	33.54	36.91	59.09	47.19	86.77	46.76	86.04	55.47
GPT-4	44.04	33.64	35.15	58.85	49.39	85.14	47.89	84.08	54.77
Mean	49.53	38.93	42.53	60.68	53.10	85.86	51.67	85.79	58.51

Table 25
Accuracy scores before the manual refinement process described in Section 4.3.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	34.83	36.85	37.71	37.18	47.91	24.79	35.50	16.26	33.88
DeepL	54.47	46.59	47.04	38.14	43.22	33.33	39.89	17.92	40.08
M2M100	8.84	9.73	12.90	10.44	16.72	6.72	18.40	5.30	11.13
M2M100$_{LG}$	13.60	15.57	17.40	15.32	19.67	13.60	20.86	9.46	15.69
MBart50	15.58	15.96	17.62	15.26	30.00	–	22.10	8.65	17.88
MBart50$_{MTM}$	16.23	17.55	18.98	12.91	28.61	–	22.85	9.62	18.11
OPUS$_{BIL}$	13.91	24.69	20.65	18.01	24.73	17.60	26.43	–	20.86
OPUS$_{MUL}$	7.84	9.86	10.58	4.33	11.27	7.69	17.79	3.25	9.08
NLLB-200$_{SM}$	18.52	25.90	21.90	21.13	29.47	17.54	26.65	12.26	21.67
NLLB-200$_{MD}$	28.86	33.95	33.92	29.62	36.88	24.51	32.92	16.48	29.64
NLLB-200$_{MDD}$	34.11	34.93	37.69	32.62	40.96	27.00	37.24	19.59	33.02
NLLB-200$_{LG}$	38.53	38.26	43.48	39.23	40.60	30.85	39.77	20.45	36.40
LLaMA2-7B	26.95	34.47	32.52	25.13	38.39	14.29	25.67	8.00	25.68
LLaMA3-8B	40.58	39.71	38.85	37.25	48.02	18.82	34.97	11.39	33.70
Mistral-7B	34.12	30.05	36.26	28.90	38.54	14.94	27.44	6.82	27.13
Gemma-2B	24.14	29.77	20.50	24.53	35.89	20.75	25.11	6.52	23.40
Gemma-7B	27.80	32.96	32.39	24.86	44.26	14.71	29.74	8.82	26.94
Gemma2-9B	49.15	48.64	53.13	52.40	61.51	30.49	45.85	21.84	45.38
Phi3-mini	34.48	33.95	33.81	21.39	32.94	11.11	24.15	3.03	24.36
Tower-7B	46.25	47.80	46.22	44.44	53.56	29.03	43.24	14.29	40.60
GPT-3.5$_{TURBO}$	50.72	52.28	50.12	46.30	60.34	28.41	49.01	18.28	44.43
GPT-4	53.91	54.11	53.74	53.31	65.36	34.34	51.88	22.64	48.66
Mean	30.61	32.44	32.61	28.76	38.58	21.03	31.70	12.42	28.52

Table 26
SFI scores for each system and language. Higher is better.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	53.24	51.20	47.88	59.73	49.82	42.91	35.51	45.42	48.21
DeepL	74.34	54.31	58.60	62.30	46.13	49.96	41.43	50.15	54.65
MBart50	25.70	22.66	26.06	27.99	29.07	–	19.35	27.20	25.43
MBart50$_{MTM}$	23.64	25.19	25.00	27.02	28.37	–	19.29	23.35	24.55
M2M100	21.38	21.20	20.77	22.62	11.84	18.33	17.00	22.19	19.42
M2M100$_{LG}$	24.25	26.33	25.09	29.25	16.50	22.37	17.92	25.45	23.39
OPUS$_{BIL}$	24.41	28.94	28.06	29.86	28.50	25.04	25.42	–	27.18
OPUS$_{MUL}$	14.38	19.35	15.80	18.03	4.91	9.35	16.19	17.48	14.44
NLLB-200$_{SM}$	30.11	32.54	34.17	39.91	28.12	27.40	22.10	30.37	30.59
NLLB-200$_{MD}$	46.20	42.80	47.21	45.40	37.10	37.41	33.54	42.11	41.47
NLLB-200$_{MDD}$	50.88	43.13	51.96	46.97	39.17	38.01	38.03	43.48	43.95
NLLB-200$_{LG}$	53.59	46.01	54.92	57.86	38.50	43.95	41.10	49.18	48.14
Llama2-7b	39.78	38.56	39.90	40.01	33.80	24.91	30.13	35.17	35.28
Llama3-8B	51.07	45.77	44.68	55.71	45.76	32.74	36.34	41.98	44.26
Mistral-7b	46.17	40.30	40.70	43.47	36.05	26.80	33.10	33.28	37.48
Gemma-2B	36.72	38.47	31.30	34.94	33.45	17.75	21.34	33.38	30.92
Gemma-7B	39.62	43.43	39.13	46.67	39.53	19.79	26.88	26.85	35.24
Gemma2-9B	63.27	54.54	62.50	74.38	64.58	57.24	52.52	54.16	60.40
Phi3-mini	51.04	44.88	40.13	37.90	30.04	15.41	22.43	21.51	32.92
Tower-7B	60.08	59.12	55.21	63.52	48.87	26.25	47.43	36.57	49.63
GPT-3.5$_{TURBO}$	66.69	61.88	59.72	68.33	61.28	47.00	53.63	58.96	59.69
GPT-4	71.45	64.99	61.43	73.53	66.64	55.31	56.74	65.08	64.40
Mean	44.00	41.16	41.37	45.70	37.18	31.90	32.16	37.30	38.71

Table 27
PDI scores for each system and language. Higher is better.

Model	DE	ES	IT	RU	ZH	BG	NL	SL	Mean
Google	58.10	54.61	54.89	65.43	54.36	49.08	40.47	52.44	53.67
DeepL	78.58	59.98	64.97	69.28	50.82	57.97	46.89	59.55	61.01
MBart50	28.23	25.85	31.94	32.31	34.10	–	20.60	36.41	29.92
MBart50$_{\text{MTM}}$	26.29	26.90	32.30	31.61	33.17	–	19.97	32.56	28.97
M2M100	21.63	22.26	22.47	24.68	15.35	19.67	15.15	27.50	21.09
M2M100$_{\text{LG}}$	25.63	27.16	27.44	32.09	21.26	25.13	18.50	30.23	25.93
OPUS$_{\text{BIL}}$	27.64	31.63	31.16	33.25	30.72	30.91	26.97	–	30.33
OPUS$_{\text{MUL}}$	16.17	19.86	18.07	17.52	6.86	10.74	15.58	20.45	15.66
NLLB-200$_{\text{SM}}$	33.52	35.95	38.62	45.44	33.18	33.31	25.23	38.34	35.45
NLLB-200$_{\text{MD}}$	49.86	47.29	51.98	52.48	41.42	44.36	36.56	49.16	46.64
NLLB-200$_{\text{MDD}}$	54.51	47.36	56.59	53.29	42.99	46.24	41.18	51.86	49.25
NLLB-200$_{\text{LG}}$	58.75	51.29	61.24	62.73	43.75	51.06	45.09	56.98	53.86
Llama2-7b	43.68	42.77	44.59	47.27	37.76	27.35	31.69	43.11	39.78
Llama3-8B	57.64	49.59	49.82	61.04	49.95	36.89	40.09	52.32	49.67
Mistral-7b	52.64	42.86	47.25	50.00	40.74	32.68	34.03	39.85	42.51
Gemma-2B	39.38	40.57	36.64	39.98	35.96	22.16	21.83	37.18	34.21
Gemma-7B	45.39	47.38	45.66	49.94	43.88	26.46	28.90	34.00	40.20
Gemma2-9B	68.04	61.22	68.35	78.06	65.30	63.32	55.73	63.26	65.41
Phi3-mini	56.22	46.78	44.86	38.33	32.52	18.88	25.47	29.42	36.56
Tower-7B	66.34	62.00	61.68	68.27	54.10	36.28	51.38	49.29	56.17
GPT-3.5$_{\text{TURBO}}$	68.78	64.89	65.75	71.33	62.05	54.59	57.37	64.86	63.70
GPT-4	74.59	69.35	67.48	76.57	68.41	62.28	61.50	70.48	68.83
Mean	47.80	44.43	46.53	50.04	40.85	37.47	34.55	44.73	43.13

Acknowledgments

Federico Martelli acknowledges the support of the CREATIVE project (CRoss-modal understanding and gEnerATIon of Visual and tExtual content, Progetti di Interesse Nazionale - PRIN 2020), which is funded by the Italian Ministry of University and Research (MUR). Tina Munda acknowledges the support of the Recovery and Resilience Plan (NOO; Načrt za okrevanje in odpornost) by the Slovenian Research and Innovation Agency (ARIS) and NextGenerationEU via the PoVeJMo research program (Adaptive Natural Language Processing with Large Language Models; Prilagodljiva obdelava naravnega jezika s pomočjo velikih jezikovnih modelov) and the financial support received from ARIS through research core funding no. P6-0411 – Language Resources and Technologies for Slovene. Svetla Koeva acknowledges the support of the project Semantic Resources and Language Processing Tools (lexical-semantic networks and language models), funded by the Bulgarian Academy of Sciences. Carole Tiberius acknowledges the support of the Leiden University Centre of Linguistics. Lastly, Roberto Navigli acknowledges the support of the PNRR MUR project PE0000013-FAIR ("Future AI Research").

References

Abdin, Marah I., Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. Technical Report MSR-TR-2024-12, Microsoft.

Ainslie, Joshua, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901. https://doi.org/10.18653/v1/2023.emnlp-main.298

Alves, Duarte Miguel, José Pombal, Nuno M. Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. In *Proceedings of First Conference on Language Modeling*.

Barba, Edoardo, Luigi Procopio, and Roberto Navigli. 2021. ConSeC: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503. https://doi.org/10.18653/v1/2021.emnlp-main.112

Bawden, Rachel and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: The case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bevilacqua, Michele, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. https://doi.org/10.24963/ijcai.2021/593

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 33:1877–1901.

Camacho-Collados, Jose and Roberto Navigli. 2017. BabelDomains: Large-scale domain labeling of lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 223–228. https://doi.org/10.18653/v1/E17-2036

Campolungo, Niccolò, Federico Martelli, Francesco Saina, and Roberto Navigli. 2022. DiBiMT: A novel benchmark for measuring word sense disambiguation biases in machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4352. https://doi.org/10.18653/v1/2022.acl-long.298

Conia, Simone and Roberto Navigli. 2021. Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics:*

Main Volume, pages 3269–3275. https://doi.org/10.18653/v1/2021.eacl-main.286

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

Devlin, Jacob, 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.

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Elfwing, Stefan, Eiji Uchibe, and Kenji Doya. 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. https://doi.org/10.1016/j.neunet.2017.12.012, PubMed: 29395652

Emelin, Denis, Ivan Titov, and Rico Sennrich. 2020. Detecting word sense disambiguation biases in machine translation for model-agnostic adversarial attacks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7635–7653. https://doi.org/10.18653/v1/2020.emnlp-main.616

Fan, Angela, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-Centric Multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Freitag, Markus, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474. https://doi.org/10.1162/tacl_a_00437

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, et al. 2024. Gemini: A family of highly capable multimodal models. Technical report, Google DeepMind.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on Gemini research and technology. Technical report, Google DeepMind.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. Technical report, Google DeepMind.

Gonzales, Annette Rios, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19. https://doi.org/10.18653/v1/W17-4702

Goyal, Naman, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538. https://doi.org/10.1162/tacl_a_00474

Graham, Yvette, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41.

Ide, Nancy and Keith Suderman. 2004. The American national corpus first release. In *Proceedings of the Fourth International*

Conference on Language Resources and Evaluation (LREC'04).

Iyer, Vivek, Edoardo Barba, Alexandra Birch, Jeff Z. Pan, and Roberto Navigli. 2023. Code-switching with word senses for pretraining in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12889–12901. https://doi.org/10.18653/v1/2023.findings-emnlp.859

Iyer, Vivek, Pinzhen Chen, and Alexandra Birch. 2023. Towards effective disambiguation for machine translation with large language models. In *Proceedings of the Eighth Conference on Machine Translation*, pages 482–495. https://doi.org/10.18653/v1/2023.wmt-1.44

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93. https://doi.org/10.1093/biomet/30.1-2.81

Kocmi, Tom, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42. https://doi.org/10.18653/v1/2023.wmt-1.1

Kocmi, Tom, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, et al. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45. https://doi.org/10.18653/v1/2023.wmt-1.1

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86.

Krovetz, Robert and W. Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)*, 10(2):115–141. https://doi.org/10.1145/146802.146810

Langone, Helen, Benjamin R. Haskell, and George A. Miller. 2004. Annotating WordNet. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 63–69.

Lee, Katherine, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. In *NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language (IRASL)*.

Lefever, Els and Veronique Hoste. 2010. SemEval-2010 Task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 15–20. https://doi.org/10.3115/1621969.1621984

Lefever, Els and Véronique Hoste. 2013. SemEval-2013 Task 10: Cross-lingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 158–166.

Ljubešić, Nikola and Kaja Dobrovoljc. 2019. What does neural bring? Analysing improvements in morphosyntactic annotation and lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 29–34. https://doi.org/10.18653/v1/W19-3704

Lommel, Arle, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica*, (12):455–463. https://doi.org/10.5565/rev/tradumatica.77

Maru, Marco, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4724–4737. https://doi.org/10.18653/v1/2022.acl-long.324

Marvin, Rebecca and Philipp Koehn. 2018. Exploring word sense disambiguation abilities of neural machine translation systems. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 125–131.

McCarthy, Diana and Roberto Navigli. 2007. SemEval-2007 Task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. https://doi.org/10.3115/1621474.1621483

Mihalcea, Rada, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 9–14.

Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41. https://doi.org/10.1145/219717.219748

Miller, George A., Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop*. https://doi.org/10.3115/1075671.1075742

Navigli, Roberto. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):1–69. https://doi.org/10.1145/1459352.1459355

Navigli, Roberto, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. https://doi.org/10.24963/ijcai.2021/620

Navigli, Roberto and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225.

Neves, Mariana, Antonio Jimeno Yepes, Amy Siu, Roland Roller, Philippe Thomas, Maika Vicente Navarro, Lana Yeganova, Dina Wiemann, Giorgio Maria Di Nunzio, Federica Vezzani, Christel Gerardin, et al. 2022. Findings of the WMT 2022 biomedical translation shared task: Monolingual clinical case reports. In *Proceedings of the Seventh Conference on Machine Translation*, pages 694–723.

Nivre, Joakim, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. GPT-4 technical report. Technical report, OpenAI.

Orlando, Riccardo, Simone Conia, Fabrizio Brignone, Francesco Cecconi, and Roberto Navigli. 2021. AMuSE-WSD: An all-in-one multilingual system for easy Word Sense Disambiguation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 298–307. https://doi.org/10.18653/v1/2021.emnlp-demo.34

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. https://doi.org/10.3115/1073083.1073135

Paul, Douglas B. and Janet M. Baker. 1992. The design for the Wall Street Journal-based CSR corpus. In *Speech and Natural Language: Proceedings of a Workshop*. https://doi.org/10.3115/1075527.1075614

Popović, Maja. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. https://doi.org/10.18653/v1/W15-3049

Popović, Maja. 2017. chrF++: Words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618. https://doi.org/10.18653/v1/W17-4770

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. https://doi.org/10.18653/v1/W18-6319

Pradhan, Sameer, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007.

SemEval-2007 Task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92. https://doi.org/10.3115/1621474.1621490

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108. https://doi.org/10.18653/v1/2020.acl-demos.14

Raganato, Alessandro, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110. https://doi.org/10.18653/v1/E17-1010

Raganato, Alessandro, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW test suite at WMT 2019: Automatically harvested multilingual contrastive word sense disambiguation test sets for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480. https://doi.org/10.18653/v1/W19-5354

Rei, Ricardo, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Rios, Annette, Mathias Müller, and Rico Sennrich. 2018. The word sense disambiguation test suite at WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 588–596. https://doi.org/10.18653/v1/W18-6437

Shazeer, Noam. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Shazeer, Noam. 2020. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Su, Jianlin, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063. https://doi.org/10.1016/j.neucom.2023.127063

Tang, Yuqing, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466. https://doi.org/10.18653/v1/2021.findings-acl.304

Tiedemann, Jörg. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218.

Tiedemann, Jörg and Santhosh Thottingal. 2020. OPUS-MT – Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 30:6000–6010.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. https://doi.org/10.1038/s41592-020-0772-5

Voita, Elena, Rico Sennrich, and Ivan Titov. 2021. Language modeling, lexical translation, reordering: The training process of NMT through the lens of classical SMT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8478–8491. https://doi.org/10.18653/v1/2021.emnlp-main.667

Xu, Haoran, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *Proceedings of The Twelfth*

International Conference on Learning Representations.

Zhang, Biao, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *Proceedings of the 40th International Conference on Machine Learning*, pages 41092–41110.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proceedings of the International Conference on Learning Representations*.

Zhu, Wenhao, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781. https://doi.org/10.18653/v1/2024.findings-naacl.176