Natural Language Processing (2026), **32**, pp. 451–469
doi:10.1017/nlp.2026.10033

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Balancing accuracy and efficiency: Evaluating encoder- and decoder-based models for word sense disambiguation and regular polysemy detection

Pin-Er Chen, Da-Chen Lian and Shu-Kai Hsieh

Graduate Institute of Linguistics, National Taiwan University, Taiwan
Corresponding author: Da-Chen Lian; Email: d08944019@ntu.edu.tw

(Received 14 January 2025 UTC; revised 23 May 2026 UTC; accepted 9 June 2026 UTC;
first published online 8 July 2026 UTC)

Abstract

This study investigates the nuanced challenges of fine-grained word sense disambiguation (WSD) tasks with regular polysemy detection (RPD) of the named entity, focusing on evaluating the trade-offs between encoder and decoder-based model performance and computational efficiency. The datasets, including Chinese Wordnet 2.0 (CWN) as sense inventory, the Social Media Corpus (PTT) for user-generated content, and the Academia Sinica Balanced Corpus (ASBC) for formal linguistic data, were chosen to provide a diverse and representative framework for evaluating both common nouns and proper nouns with regular polysemy in Taiwan Mandarin. This analysis evaluated ten encoder- and decoder-based models, assessing their performance on two tasks. The encoder-based models demonstrate comparable accuracy to the decoder-based models on WSD tasks (77.5% vs. 78.5%), and similarly strong performance in RPD tasks (84.2% vs. 83.8%). On a large-scale all-words WSD task, the encoder model not only outperformed the decoder model but also generated substantially lower carbon emissions – an eight-fold reduction. These differences underscore the trade-offs between model architecture and task-specific performance, highlighting the necessity for balancing performance and energy efficiency in the design and application of language models, advocating for sustainable and eco-friendly practices in natural language processing development.

Keywords: Word sense disambiguation; regular polysemy detection; sustainability; efficiency; encoder vs. decoder

1. Introduction

Natural language processing (NLP) has witnessed remarkable advancements in recent years, driven by the development of increasingly sophisticated machine learning models. Among these, encoder-based architectures, such as BERT (Devlin *et al.* 2019), and decoder-based architectures, like GPT (Brown *et al.* 2020), have emerged as two dominant paradigms. While both encoder- and decoder-based models have achieved significant progress in NLP, questions persist about their relative strengths, particularly regarding performance, computational efficiency, and energy consumption.

Evaluating the performance and sustainability of NLP models is therefore crucial. As encoder- and decoder-based models continue to evolve, so do their computational demands and environmental footprints. Previous research, such as Hsieh's exploration of fine-grained word sense disambiguation (WSD) and regular polysemy detection (RPD) (Hsieh *et al.* 2024), provides a foundation for investigating the capacities of encoder-based models on these tasks. However,

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

understanding how different architectures balance accuracy, efficiency, and energy consumption when applied to disambiguation tasks remains an open question.

This study builds on Hsieh *et al.* (2024) in three substantive directions. First, the prior study evaluated a single encoder-based architecture, a GlossBERT (Huang *et al.* 2019) model fine-tuned on Chinese Wordnet 2.0 (CWN)[a] glosses, as a unified disambiguator for common nouns and regular polysemy of proper nouns. The present work extends this to a systematic comparison of ten models: seven encoder-based architectures spanning BERT and multiple DeBERTa variants (97.7M to 435 M parameters) and three decoder-based architectures (SmolLM-180 M, Gemma-2-2b, and Llama-3.2-3B; 177 M to 3.21B parameters). This encoder–decoder comparison on WSD and RPD tasks in Taiwan Mandarin has not previously been undertaken. Second, Hsieh *et al.* (2024) evaluated models exclusively on held-out test sets drawn from the same CWN and PTT[b] sources; the present work adds a large-scale all-words sense tagging experiment on the Academia Sinica Balanced Corpus (ASBC),[c] comprising 32.7 million instances, which directly tests how each architecture scales beyond curated evaluation sets and enables the broader development of Taiwan Mandarin lexical resources. Third, and distinctively, Hsieh *et al.* (2024) did not address the computational cost of model training or inference. The present study introduces training energy consumption, measured via NVML hardware counters and converted to CO_2e following Patterson *et al.* (2022) and Luccioni *et al.* (2023), as an explicit evaluation dimension alongside task accuracy, situating model selection within the broader discourse on sustainable NLP.

This paper is organized as follows. Section 2 reviews related work in the fields of WSD and RPD, highlighting key advancements and existing challenges. Section 3 introduces the datasets and model configurations used in this study, including detailed discussions of the datasets, model inputs, and specifications for the models and the training process. Section 4 presents the results, providing an evaluation of the models on both WSD and RPD tasks. Section 5 focuses on the all-words word sense tagging task, offering insights into large-scale sense disambiguation, model configuration, and a comprehensive analysis of the results. Finally, Section 6 concludes the paper by summarizing key findings and suggesting potential directions for future work in sustainable NLP.

2. Related work

WSD has been a fundamental challenge in computational linguistics for decades. Traditional approaches can be broadly categorized into two main paradigms: knowledge-based and supervised learning methods. Knowledge-based techniques utilize lexical resources and semantic networks, employing various similarity metrics and graph-based algorithms to determine word senses (Agirre, de Lacalle, and Soroa 2014; Scozzafava *et al.* 2020; Wang and Wang 2020). In contrast, supervised approaches leverage machine learning algorithms trained on annotated corpora, with recent advances in neural architectures significantly improving performance. The introduction of contextual embeddings and transformer-based models has led to notable breakthroughs in the performance of WSD tasks (Peters *et al.* 2018; Devlin *et al.* 2019; Liu *et al.* 2019; Blevins and Zettlemoyer 2020; Conia and Navigli 2021; Wang and Wang 2021).

Recent architectural innovations have explored novel frameworks for sense disambiguation. Notable examples include sequence classification approaches that utilize glosses and span extraction methodologies (Huang *et al.* 2019; Barba, Pasini, and Navigli 2021). A paradigm shift has occurred with the emergence of large language models (LLMs), which have transformed

[a]Chinese Wordnet 2.0 (hereafter, CWN) is a computational lexical resource designed to capture the semantic structure of Taiwan Mandarin. It comprises a network of synsets that represents groups of synonymous words, linked by semantic relations. https://lopentu.github.io/CwnWeb/

[b]The PTT Corpus is a large-scale collection of Mandarin social media posts organized by topic-based boards. The 20 selected boards include Horror, Hate, Marvel, WomenTalk, Gossiping, JapanMovie, HatePolitics, Tainan, LGBT, SEX, BabyMother, NTU, nCoV2019, Data, Sex, Food, Kaohsiung, Marriage, Gay, Boy-Girl, and Joke.

[c]https://asbc.iis.sinica.edu.tw/

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

the traditional classification-based approach to a generative one. Recent studies have shown impressive results with instruction-tuned models such as GPT-4 (OpenAI *et al.* 2024), Llama-3-70B (Shi *et al.* 2024), and PaLM-2 (Anil *et al.* 2023) achieving state-of-the-art performance on multiple-choice, true/false, or zero-shot evaluations (Yae *et al.* 2024; Rajpoot, Jindal, and Parikh 2024). Furthermore, multilingual evaluations of LLMs in specialized domains (e.g., clinical, biomedical, or economic) have shown promising results for English content while highlighting challenges in other languages (Kang, Blevins, and Zettlemoyer 2024; Hosseini, Hosseini, and Javidan 2024; Kugic, Schulz, and Kreuzthaler 2024). It should be noted that WSD intersects with broader questions of lexical semantics, particularly in distinguishing between regular polysemy, where sense alternations follow systematic patterns across words, and irregular cases requiring explicit disambiguation (Boleda, Schulte im Walde, and Badia 2012). This theoretical foundation continues to inform modern approaches to WSD.

Regular polysemy refers to the systematic coexistence of multiple related meanings within a single word, following shared semantic patterns across similar terms (Apresjan, 1974). Computational approaches to modeling regular polysemy have included decision trees, embedding techniques, and distributional models (Boleda, Schulte im Walde, and Badia 2012; Del Tredici and Bel 2015; Chersoni, Lenci, and Blache 2017). Although these methods address key aspects of polysemy, they often overlook proper name ambiguity. To formalize polysemous types, this study follows the methodology of Hsieh *et al.* (2024), including Apresjan (1974)'s definition for regular polysemy and Pustejovsky (1995)'s dot-type framework, which has been widely applied for systematic representation. Within this framework, a **dot object** is a theoretical construct encoding the systematic polysemous relationships of a word or entity. For instance, the proper noun *Starbucks* can simultaneously invoke the meanings of a producer, a product, and a location – represented as the dot object *Producer·Product·Location*. In computational tasks, each dot object is operationalized through **type classes**: context-specific labels assigned to a target word based on its meaning in a given sentence. For example, *Starbucks* in the sentence "I grabbed a coffee from Starbucks" would be assigned the type class *Location*, whereas in "Starbucks released a new blend" it would be assigned *Product*. This distinction between dot objects (the theoretical framework) and type classes (the operational labels) is central to the RPD task explored in this study and is elaborated further in Section 3.2.

3. Dataset and model configuration

The experimental framework in this study adopts the datasets introduced by Hsieh *et al.* (2024) but substantially extends the model evaluation scope. Detailed descriptions of the annotation and construction process of the datasets are available in the original study. While Hsieh *et al.* (2024) trained and assessed a single encoder-based architecture (DottedWSD) on WSD and RPD tasks, the present work evaluates ten models spanning both encoder-based architectures and decoder-based architectures. In addition to task-specific accuracy, we record training time, GPU energy consumption, and estimated carbon emissions for every model run, introducing computational sustainability as an explicit evaluation criterion alongside performance. The following subsections describe the datasets (Sections 3.1 and 3.2), the input construction for both model types (Section 3.3), and the model configurations and training protocol, including carbon estimation methodology (Section 3.4). The code and model configurations used in this study are publicly available.[d]

3.1 WSD dataset

The WSD dataset includes sentences with one target word, annotated with its sense and part-of-speech, leveraging examples from the Chinese Wordnet 2.0 (CWN); Table 1 shows an example

[d]https://github.com/lopentu/dotted-wsd-public

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 1. Example sense structure of CWN with translation

Field	Content	Translation
Lemma	永遠 _1	Forever_1
POS	Nd	Nd
Definition	長久持續的時間。	A long-lasting period of time.
Example sentence	希望我們把共同的關懷之心繼續散布，直到 < 永遠 >!	I hope we continue spreading our shared care until < forever> !
Relation	synonym, <(永久, Nd): 長久持續的時間。 >	synonym, < (Permanent, Nd): A long-lasting period of time.>

Table 2. Dataset statistics. The sequences are derived from the example sentences

Dataset	Train split	Test split	Total
WSD example sentences	36,622	9,162	**45,784**
RPD example sentences	3,641	866	**4,507**
WSD sequences	386,351	83,418	469,769
RPD sequences	9,227	2,333	11,560

of the sense structure in CWN. The sense inventory of CWN provides fine-grained distinctions, with some words having more than 20 senses, yet the example sentences in CWN may fail to fully represent collocation environments for these complex words. To address this, manual sense annotation was performed in Hsieh *et al.* (2024) for 113 words with more than 10 senses, using 27,012 sentences extracted from the ASBC. Sentences were filtered to exclude certain uses, such as proper nouns, transliterations, or segmentation errors. Six trained annotators, using a web-based tool, categorized the senses into hierarchical trees with an average depth of 2.77. The final annotation dataset comprises 28,836 examples in which the frequently used senses dominate, and only 57% of the senses are observed. Combined with CWN-derived data, the WSD dataset totals 45,784 examples, shown in the first row of Table 2. The dataset has a skewed sense distribution (see Appendix A), with more frequent senses making up a substantial portion of the data, ensuring a comprehensive representation of common polysemy.

3.2 RPD dataset

First, we differentiate between the concepts of *dot object* and *type class* to provide clarity on their roles in modeling regular polysemy. A *dot object* is a theoretical construct that represents systematic polysemous relationships in language, showing how a word or entity can carry multiple interrelated meanings. For example, the dot object "Producer.Product.Location reflects the different meanings of Starbucks" and alike, which can refer to an organization, a product, or a location. A *type class*, however, is a practical, context-specific application of a dot object that serves as a categorical label used in computational tasks to annotate and disambiguate the meaning of a term in its given context. For instance, "Starbucks" may be labeled as one of the three type classes (i.e., `producer`, `product`, or `location`) depending on its context. While dot objects offer the theoretical framework, type classes apply this framework in computational tasks.

The RPD dataset focuses on **proper nouns** identified as regularly polysemous within the corpus. It contains 4,507 example sentences sourced from the PTT Corpus, as indicated in the second

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 3. Dot objects mapped to Wikidata categories with the number of proper nouns in the dataset

Dot object	Wikidata category	Number of proper nouns
Info*Phy	Work of art	4
Loc*Org	Human-geographic territorial entity, government agency, industrial zone, intergovernmental organization, court, Chinese temple, museum	187
Org*Hum	Political party, military unit, sports organization, religious organization, religious identity	39
Org*Info*Phy*Hum	Mass media	22
Org*Loc*Hum	University, educational institution, organization, hospital	31
Phy*Evt*Hum	Award	7
Prcr*Prct*Loc	Business	57

Table 4. Regular polysemy type class with Mandarin translation and its gloss

RP type class	Translation	Gloss
Physical	有形的	Tangible objects.
Organization	機構	General reference to administrative and functional structures
Producer	作者; 制造商	Creators of poetry, articles, or other artworks; businesses that produce or supply goods or services
Product	作品	Artistic creations; commodities that have been produced
Location	地點	Positions or occupied sites
Event	事件	Circumstances, incidents
Human	人類	General term for humanity
Information	資訊	General reference to data, knowledge, and messages

row of Table 2. In total, 347 proper nouns are included, each appearing in one or more sentences. The proper nouns are annotated with one of seven predefined dot objects,[e] which capture systematic sense alternations observed in these entities. Table 3 presents the seven dot objects and their corresponding semantic categories, aligned with Wikidata entries as detailed by Hsieh *et al.* (2024). Each instance of a proper noun in a specific sentence is assigned a type class based on its contextual meaning. The type classes involved in this dataset are shown in Table 4.

3.3 Model input

In the Transformer architecture, encoder-based models focus mainly on the encoder component, which is designed to process and encode input text into rich contextual representations by capturing bidirectional dependencies. These models are called encoder-based because they emphasize understanding and transforming the input into a meaningful representation for tasks

[e]The seven dot objects predefined in Hsieh *et al.* (2024)'s RPD dataset: information.physical (Info.Phy), location.organization (Loc.Org), organization.human (Org.Hum), organization.information.physical.human (Org.Info.Phy.Hum), organization.location.human (Org.Loc.Hum), physical.event.human (Phy.Evt.Hum), producer.product.location (Prcr.Prct.Loc)

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 5. The three model input sequences generated for 狀 *zhuàng* (“case”) for WSD. Each instance is represented with its translation below; only the instance in Taiwan Mandarin is used as the model input. When a POS hint is provided (as in the ”By example (POS)” evaluation), candidate senses whose POS tag does not match the target word’s annotated POS are excluded from the input sequences, resulting in fewer sequences per test sentence

[CLS] 雇主一〈狀〉告到上頭… [SEP] 狀: 構成特定對象的外部輪廓… [SEP] False
[CLS] The employer filed a <case> to the higher authorities [SEP] case: The external outline of a specific object [SEP] False
[CLS] 雇主一〈狀〉告到上頭… [SEP] 狀: 請求司法機關審理的案件… [SEP] False
[CLS] The employer filed a <case> to the higher authorities [SEP] case: A legal matter submitted to judicial authorities for adjudication [SEP] False
[CLS] 雇主一〈狀〉告到上頭… [SEP] 狀: 刻意向較有權力的人… [SEP] True
[CLS] The employer filed a <case> to the higher authorities [SEP] case: Deliberate appeal to a more powerful entity [SEP] True

like classification, named entity recognition, and question answering. On the other hand, decoder-based models utilize only the decoder component of the Transformer. The decoder is designed for autoregressive processing, generating text sequentially while attending to previous outputs, making these models ideal for tasks like text generation, summarization, and translation.

For the preparation of input data for different model types, we construct the input sequences of the models as context–gloss pairs. For *encoder-based* models, each test sentence t_i in the WSD dataset contains a predefined target word w_i, which is linked to a sense inventory provided by CWN. For each t_i, we create j input sequences, where j represents the number of senses of w_i (i.e., each input sequence corresponds to a specific sense s_{ij} of the target word). The sense s_{ij} is described using its definition (SENSE-DEF) and an example sentence (SENSE-EX-SENT) from CWN. These components are combined with the original test sentence t_i to construct the final input sequence as follows:

$$\texttt{[CLS]}\ \overbrace{\texttt{TEST-SENT}}^{t_i}\ \texttt{[SEP]}\ \overbrace{\texttt{TGT}}^{w_i}\texttt{:}\underbrace{\texttt{SENSE-DEF, SENSE-EX-SENT}}_{s_{ij}}\ \texttt{[SEP]}$$

Thus, each test sentence generates as many sequences as there are senses in the inventory, with only one labeled correct (True). Target words having fewer than two predefined senses are excluded. For example, if a target word 狀 *zhuàng* (“case”) has three candidate senses, three sequences will be generated, as shown in Table 5. The total number of generated sequences is 469,769 as shown in the third row of Table 2. When POS hints are available, they are used as a filtering mechanism to constrain the candidate sense inventory prior to classification. Specifically, only candidate senses whose POS tag matches the annotated POS of the target word are retained as valid candidates. This reduces the number of input sequences generated per test sentence and eliminates cross-POS sense confusion. When no POS hint is provided, all senses in the inventory are included as candidates regardless of their POS tag.

For the RPD dataset, the input sequences are constructed in a similar way. Each test sentence t_i contains a single target word w_i, with its predefined RP-class inventory $rpclass_inv_i$ retrieved. For each type class $c_{ij} \in rpclass_inv_i$, its corresponding gloss (RPCLASS-GLOSS) is derived from the Revised Mandarin Chinese Dictionary. The input sequence is then concatenated as:

$$\texttt{[CLS]}\ \overbrace{\texttt{TEST-SENT}}^{t_i}\ \texttt{[SEP]}\ \overbrace{\texttt{TGT}}^{w_i}\texttt{:}\underbrace{\texttt{RPCLASS, RPCLASS-GLOSS}}_{c_{ij}}\ \texttt{[SEP]}$$

Each test sentence generates as many sequences as there are type classes in the inventory, with only one labeled correct (True). The target words are marked with angular brackets (i.e., <>) as a weak supervision signal. For example, if a target word 哈佛 *hā fó* (“Harvard”) has three candidate type classes, three sequences will be generated, as shown in Table 6. The total number of the generated sequences is 11,560, as shown in the last row in Table 2.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 6. The three model input sequences generated for 哈佛 *hā fó* ("Harvard") for RPD. Each instance is represented with its translation below; only the instance in Taiwan Mandarin is used as the model input

[CLS] 他最近爲了〈哈佛〉學費... [SEP] 哈佛: 機構, 泛指機關團體 [SEP] True
[CLS] He recently worked hard for <Harvard> tuition... [SEP] Harvard: An institution, generally referring to an organization or group [SEP] True
[CLS] 他最近爲了〈哈佛〉學費... [SEP] 哈佛: 地點, 所在的地方 [SEP] False
[CLS] He recently worked hard for <Harvard> tuition... [SEP] Harvard: A place, referring to a specific location [SEP] False
[CLS] 他最近爲了〈哈佛〉學費... [SEP] 哈佛: 人類, 人的總稱 [SEP] False
[CLS] He recently worked hard for <Harvard> tuition... [SEP] Harvard: Humanity, referring to all people collectively [SEP] False

For *decoder-based* models, the input is slightly different because [CLS] and [SEP] tokens are not natively used. However, to more closely match the input used by encoder-based models, we designate other tokens to perform equivalent roles. To construct the input for a decoder-based model, we use the predefined beginning-of-sequence ([BOS]) token for the respective model to signal the beginning of an input and to serve as the delimiter token that separates the first sequence from the second. We then utilize an unused token from the vocabulary, such as < unused99 > (token ID 255199) for Gemma-2-2b, to serve as the [CLS] token, indicating the end of the input sequence and providing the representation for the classification head. The input for the WSD task is structured as follows:

$$\texttt{[BOS]}\ \overbrace{\texttt{TEST-SENT}}^{t_i}\ \texttt{[BOS]}\ \overbrace{\texttt{TGT}}^{w_i}\texttt{:}\underbrace{\texttt{SENSE-DEF, SENSE-EX-SENT}}_{s_{ij}}\ \texttt{[CLS]}$$

Similarly, the input for the RPD task is structured as follows:

$$\texttt{[BOS]}\ \overbrace{\texttt{TEST-SENT}}^{t_i}\ \texttt{[BOS]}\ \overbrace{\texttt{TGT}}^{w_i}\texttt{:}\underbrace{\texttt{RPCLASS, RPCLASS-GLOSS}}_{c_{ij}}\ \texttt{[CLS]}$$

The actual tokens used for each decoder-based model can be found in Appendix B.

It is worth noting a limitation of the current input design. By appending the target word and candidate sense definition after the test sentence, both encoder- and decoder-based models have access to the full sentential context (both preceding and subsequent). This effectively neutralizes one of the key architectural differences between the two model types: while encoders are inherently bidirectional, decoders are designed to attend only to preceding context. The current setup inadvertently grants decoder-based models access to subsequent context, providing them with an additional advantage that would not exist in a purely autoregressive setting.

3.4 Models and training details

3.4.1 Models

To evaluate the impact of model architecture on performance, we select a diverse range of encoder-based models, from Google's original BERT (Devlin *et al.* 2019) to newer architectures, such as Microsoft's *DeBERTa* (He, Gao, and Chen 2023). The encoder-based models range in size from 97.7M to 435 M parameters, while the decoder-based models range from 177 M to 3.21B parameters. The largest decoder-based model is more than an order of magnitude larger than the smallest encoder-based model. A list of all models used can be found in Appendix C. To enhance readability and facilitate discussion, we will refer to these models using their abbreviated names[f] in all subsequent sections. To use both encoder- and decoder-based models for classification,

[f]The abbreviated names for models: DottedWSD (DottedWSD), Erlangshen-DBT-v2 (Erlangshen-DeBERTa-v2-97M-Chinese), mDBT-v3-xnli (mDeBERTa-v3-base-xnli-multilingual-nli-2mil7), CkipLab-BERT (ckiplab-bert-base-chinese), DBT-v3-large (deberta-v3-large), yentinglin-BERT (yentinglin-bert-base-zhtw), Google-BERT (google-bert-bert-base-chinese), SmolLM-180 M (SmolLM-Chinese-180M), Gemma-2-2b (gemma-2-2b), and Llama-3.2-3B (meta-llama-Llama-3.2-3B).

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

we utilize Transformers' (Wolf *et al.* 2020) `AutoModelForSequenceClassification`, which allows us to easily add a classifier head with two output neurons, corresponding to the `False` and `True` classes. The respective `[CLS]` token for each model is fed into the classifier head to make a prediction.

3.4.2 Training details

We fine-tune all models using the Hugging Face Transformers (v. `4.46.3`) Trainer. We perform full-parameter fine-tuning for all encoder-based models and the `SmolLM-180M` decoder-based model. For the larger decoder-based models, `Gemma-2-2b` and `Llama-3.2-3B`, we use Low-Rank Adaptation (LoRA) (Hu *et al.* 2022) due to their size and memory requirements. The hyperparameters for full-parameter and LoRA fine-tuning can be found in Appendix D. All training was conducted on a single GPU with 24 GB of VRAM (either an Nvidia RTX 4090 or an Nvidia RTX A5000). We train for three epochs, saving a checkpoint after each epoch and select the checkpoint with the lowest cross-entropy evaluation loss.

3.4.3 Training times

The BERT base models had the shortest training times, all requiring approximately 50 minutes. `Gemma-2-2b` took the longest at 19 hours and 16 minutes, which is 23 times longer than the BERT models. Even the smallest decoder-based model, `SmolLM-180M` (177M parameters), required 6 hours and 19 minutes to train. In comparison, the encoder-based model with a similar number of parameters, `deberta-v3-base` (184M parameters), took only 2 hours and 9 minutes. Despite having more parameters, the encoder-based model trained in almost one-third of the time.

3.4.4 Carbon emission estimation details

Carbon emissions were estimated following Patterson *et al.* (2022), with an explicit system overhead factor to separate GPU-measured energy from total server energy following Luccioni *et al.* (2023). GPU energy was measured via Weights & Biases using NVML hardware counters. A system overhead factor of 1.5× was applied to account for non-GPU server power (CPU, memory, storage, fans), with PUE of 1.2 for on-premises infrastructure (NTU server room, Taiwan) and 1.3 for cloud infrastructure (RunPod, datacenter region unknown). Carbon intensity was 0.474 $kgCO_2eq/kWh$ for on-premises compute (Energy Administration, Ministry of Economic Affairs, 2024) and 0.475 $kgCO_2eq/kWh$ (conservative upper bound) for cloud compute. Results are reported in Table 10; for a comparative review of carbon estimation approaches, see Bouza *et al.* (2023).

4. Results

4.1 Model performance on WSD

Table 7 shows the evaluation of the encoder- and decoder-based models on the WSD dataset. `DottedWSD`, constructed in the previous study, emerges as a strong baseline, achieving an accuracy of 82% by instance and 86% by example with POS hints.

`Llama-3.2-3B` achieves the highest accuracy by instance (.864), outperforming the other models. In particular, it also excels in by-example evaluations with POS (.821) and without POS (.811), closely followed by the performance of `Google-BERT` by instance (.861) and by example (.818 and .813). Notably, `Llama-3.2-3B` is a decoder-based model, while `Google-BERT` is an encoder-based model.

Lower-performing models, such as `yentinglin-BERT` (.656) and `SmolLM-180M` (.770), highlight the challenges faced by models with limited scale or pretraining resources. The performance gap underscores the importance of incorporating high-quality large-scale data that capture linguistic nuances and domain-specific lexical variations.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 7. Model accuracy in WSD. Bold numbers indicate the best accuracy, while numbers with a double underline indicate the second-best accuracy. For clarity, only the best model among the same family is listed; for example, DBT-v3-large is listed among the deberta-v3 family. The accuracy of all models is shown in Figure 1

Model	By instance	By example (POS)	By example (no POS)
DottedWSD	.820	**.860**	**.820**
Erlangshen-DBT-v2	.791	.744	.712
mDBT-v3-xnli	.840	.797	.778
CkipLab-BERT	.850	.809	.798
DBT-v3-large	.829	.788	.762
yentinglin-BERT	.656	.611	.521
Google-BERT	<u>.861</u>	.819	<u>.813</u>
SmolLM-180 M	.770	.715	.662
Gemma-2-2b	.845	.804	.779
Llama-3.2-3B	**.864**	<u>.822</u>	.811

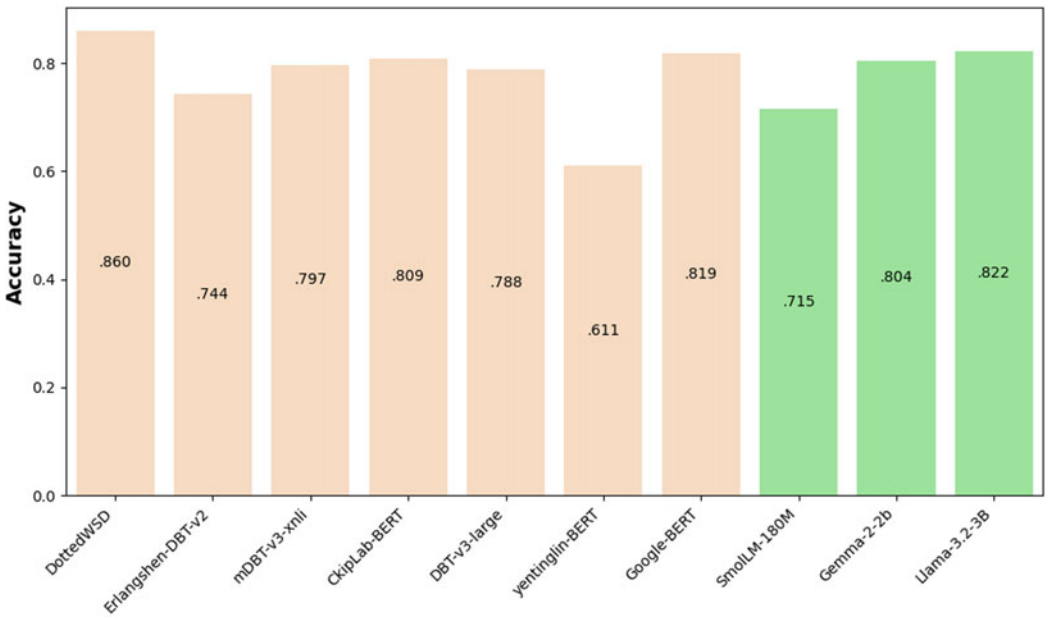


Figure 1. Model accuracy on WSD (by example) with POS hints. The yellow bars represent the encoder-based models and the green bars represent the decoder-based models.

The comparison between Figures 1 and 2 illustrates that the introduction of POS hints generally enhances performance in most models, highlighting their role in providing additional syntactic context. This aligns with the findings of Hsieh *et al.* (2024), which suggest that while POS hints yield modest improvements, the models primarily rely on distinguishing lexical semantics rather than heavily depending on POS tags. The trend is more obvious for mid-performing models, such as `SmolLM-180M`, which benefit significantly from the POS hints. In contrast, high-performing

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

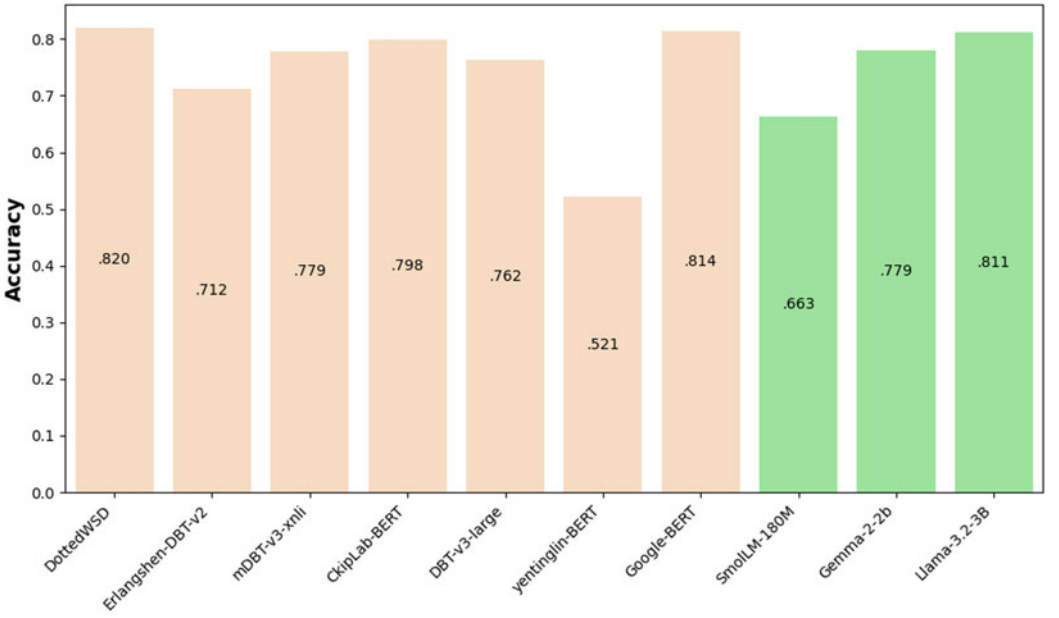


Figure 2. Model accuracy on WSD (by example) without POS hints. The yellow bars represent the encoder-based models and the green bars represent the decoder-based models.

models, such as `Google-BERT` and `Llama-3.2-3B`, demonstrate marginal improvements, suggesting that these models may already integrate sufficient contextual and syntactic information during prediction.

A particularly revealing comparison emerges between SmolLM-180 M (177M parameters) and deberta-v3-base (184M parameters), two models of comparable size but different architectures. Despite having a similar parameter count, deberta-v3-base achieves a substantially higher accuracy by instance (.840) compared to SmolLM-180 M (.770), representing a gap of 7 percentage points. This suggests that the encoder architecture holds an inherent advantage over the decoder architecture when model size is controlled for. In light of this, the comparable performance of Llama-3.2-3B (.864) relative to Google-BERT (.861) should be interpreted with caution – the similarity in their scores likely reflects the decoder model's significantly larger parameter count (3.21B vs. 103M) and more extensive pretraining data, rather than architectural equivalence. These findings further reinforce the argument that encoder-based models can achieve competitive performance at a substantially lower model scale.

When interpreting these results, it should be noted that the input design described in Section 3.3 may partially favor decoder-based models. Since the candidate sense definitions are appended after the test sentence, decoders can attend to the full context rather than only the preceding tokens. This means the performance gap between encoder- and decoder-based models may be underestimated, and encoder-based models may in fact hold a greater advantage under a more naturalistic autoregressive setting.

4.2 Model performance on RPD task

Table 8 shows the evaluation of the model accuracy when disambiguating regular polysemy. The results highlight distinctions in how encoder- and decoder-based models handle semantic alternation in proper nouns.

`Google-BERT` outperforms other models with an average accuracy of .869, indicating its robust handling of systematic sense alternations such as Location*Organization and Organization*Human. This superior performance may be attributed to its extensive multilingual

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 8. Model accuracy on RPD task. The type classes are abbreviated with their initials: I: Information, Ph: Physical, L: Location, H: Human, O: Organization, E: Event, Pr: Producer, Pt: Product. The bold number indicates the best accuracy, while the number with a double underline indicates the second-best accuracy

Model	I*Ph	L*O	O*H	O*I*Ph*H	O*L*H	Ph*E*H	Pr*Pt*L	Avg.
DottedWSD	.880	.900	.880	.760	.910	.780	.890	.857
Erlangshen-DBT-v2	.941	.884	.827	.759	.887	.634	.832	.837
mDBT-v3-xnli	.882	.905	.827	.759	.887	.732	.858	.850
CkipLab-BERT	.941	.888	.840	.771	.915	.780	.875	<u>.861</u>
DBT-v3-large	.941	.901	.853	.747	.901	.707	.858	.858
yentinglin-BERT	.765	.720	.827	.759	.872	.610	.741	.765
Google-BERT	.941	.862	.880	.795	.922	.829	.853	**.869**
SmolLM-Chinese-180M	.941	.858	.827	.759	.901	.659	.802	.841
Gemma-2-2b	.941	.884	.840	.735	.887	.683	.845	.845
Llama-3.2-3B	.824	.875	.880	.747	.901	.732	.836	.828

Table 9. Comparison of average model accuracy on WSD and RPD between encoder-based and decoder-based models

Model type	WSD				RPD
	By instance	By example (POS)	By example (no POS)	Avg.	Avg.
Encoders	.807	.775	.743	.775	.842
Decoders	.826	.780	.751	.785	.838

and domain-specific pretraining. Interestingly, `CkipLab-BERT`, trained by the CKIP Lab located in Taiwan, achieves an average accuracy of .861, outperforming several larger models. This result underscores the importance of pretraining strategies tailored to the linguistic and cultural nuances of the target language, especially for tasks that involve named entities with culture-specific usage patterns.

The variation in performance across dot-type categories, such as Information*Physical and Producer*Product*Location, suggests that the models are sensitive to the complexity of class overlaps. For example, models generally struggle more with overlapping categories like Physical*Event*Human compared to more distinct classes like Location*Organization, where contextual cues may be clearer.

In summary, Table 9 illustrates the comparable performance of encoder- and decoder-based models on WSD and RPD tasks. The average WSD accuracy is closely aligned, with encoders achieving 77.5% and decoder-based models performing slightly better at 78.5%. For the RPD task, encoder-based models achieve a higher average score of 84.2 %, compared to 83.8 % for decoder-based models. However, these aggregate figures obscure an important architectural difference. The near-equivalent performance of the two model types is largely driven by the inclusion of Llama-3.2-3B, a model with 3.21B parameters – over 31 times larger than the best-performing encoder model, Google-BERT (103M parameters). When comparing models of similar scale, such as SmolLM-180 M (177M parameters) and deberta-v3-base (184M parameters), the encoder-based model clearly outperforms its decoder counterpart (84.0 % vs. 77.0 % by instance). This size-controlled comparison strengthens the case for encoder-based models as a more parameter-efficient choice for WSD and RPD tasks and provides a basis for further exploration of their efficiency and sustainability.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 10. Training carbon emissions for all 13 runs (1 run = WSD + RPD task). CO_2e is computed following the method described in Section 3.4.4, with $f_{sys} = 1.5$, PUE of 1.2 (on-premises) and 1.3 (cloud), and carbon intensity of 0.474 $kgCO_2eq/kWh$ (on-premises) and 0.475 $kgCO_2eq/kWh$ (cloud). GPU energy is derived from NVML power samples logged via Weights & Biases. Type: E = encoder, D = decoder. † = trained with LoRA

Model	Type	Infra	GPU	GPU energy (kWh)	Runtime (h)	CO_2e (g)	%
Llama-3.2-3B†	D	Cloud	RTX 4090	4.300	12.80	3,983	28.1
Gemma-2-2b†	D	On-prem	RTX A5000	4.357	19.27	3,717	26.3
DBT-v3-large	E	Cloud	RTX 4090	1.571	5.10	1,455	10.3
SmolLM-180 M	D	On-prem	RTX A5000	1.415	6.33	1,207	8.5
Erlangshen-DBT-v2	E	Cloud	RTX 4090	0.668	2.22	619	4.4
deberta-v3-base	E	Cloud	RTX 4090	0.648	2.15	600	4.2
mDeBERTa-v3-mnli-xnli	E	On-prem	RTX A5000	0.689	3.07	588	4.2
mDBT-v3-xnli	E	Cloud	RTX 4090	0.531	1.77	492	3.5
mdeberta-v3-base	E	Cloud	RTX 4090	0.525	1.75	486	3.4
deberta-v3-small	E	Cloud	RTX 4090	0.354	1.20	328	2.3
Google-BERT	E	Cloud	RTX 4090	0.277	0.84	257	1.8
yentinglin-BERT	E	Cloud	RTX 4090	0.276	0.84	256	1.8
CkipLab-BERT	E	Cloud	RTX 4090	0.186	0.84	172	1.2
Total				**15.797**	**58.2**	**14,160**	**100**

4.3 Training carbon emissions by architecture

Table 10 reports training energy consumption and estimated CO_2e emissions for all 13 runs (1 run = WSD + RPD task). In total, training consumed 15.8 kWh of GPU energy across 58.2 GPU-hours, producing an estimated 14.2 kg CO_2e under the central scenario (see Section 3.4.4). A sensitivity analysis over overhead assumptions yields a lower bound of 7.5 kg (GPU energy only, no system or facility overhead) and an upper bound of 17.5 kg (HPC-scale $f_{sys} = 1.85$, as reported for the BLOOM training cluster). The three decoder-based models (Llama-3.2-3B, Gemma-2-2b, SmolLM-180M) accounted for 62.9% of total emissions despite comprising only 3 of 13 runs. This reflects the substantially higher compute cost of autoregressive fine-tuning: decoder models generate output token-by-token and must process the full sequence at each step, whereas encoder models perform a single forward pass over the entire input. The emissions asymmetry is particularly evident in the size-controlled comparison between SmolLM-180 M (177M parameters, decoder) and deberta-v3-base (184M parameters, encoder): despite near-identical parameter counts, SmolLM-180 M produced 1,207 g CO_2e compared to 600 g for deberta-v3-base – less than half the emissions for a comparable model scale.

5. Large-scale all-words sense tagging task

Building on the results of our initial experiment, which has revealed that encoder- and decoder-based models exhibit comparable performance on WSD and RPD tasks, as illustrated in Table 9, we investigate the efficiency and sustainability of these models further, particularly regarding energy consumption and carbon emissions. We conduct a second experiment centered on large-scale, all-words sense tagging using data from the ASBC. This experiment focuses on two high-performing models, `Google-BERT` and `Llama-3.2-3B`, to perform a detailed evaluation of their sense tagging capabilities. The aim is to assess their scalability on larger datasets and

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table 11. Comparison of model accuracy, GPU resources, total GPU hours, and estimated carbon emissions produced on the all-words sense tagging task. Carbon emissions follow the estimation method in Section 3.4.4; the A40 inference run uses the cloud PUE and carbon-intensity values defined there.

Model	Type	GPU (total RAM)	GPU hours	Accuracy (%)	Carbon emissions (kg of CO_2 eq.)
Google-BERT	E	1 x A5000 (24 GB)	17	94.5	1.69
Llama-3.2-3B	D	6 x A40 (288 GB)	126	92.5	13.61

explore the implications for designing environmentally sustainable models while enhancing the accessibility of lexical resources for wider use.

5.1 Data and model configuration

Each line within the ASBC is processed to identify words with multiple potential senses according to their corresponding sense inventory. This results in a total of 32.7 million instances. The transformation of the model input follows the procedure for creating training and testing data in Section 3.3, where the complete sentence containing the target word `TGT` is represented as `TEST-SENT`, the sense definition as `SENSE-DEF`, and one example sentence for the given sense as `SENSE-EX-SENT`. The input structure of the encoder-based models is defined as:

`[CLS] TEST-SENT [SEP] TGT, SENSE-DEF, SENSE-EX-SENT [SEP]`

In contrast, the decoder-based model input structure is given as:

`[BOS] TEST-SENT [BOS] TGT, SENSE-DEF, SENSE-EX-SENT [CLS]`

Model inference was conducted using both `Google-BERT` and `Llama-3.2-3B`. `Google-BERT` inference is performed on a single Nvidia A5000 GPU with 24 GB of VRAM, requiring approximately 17 hours. To expedite the process for `Llama-3.2-3B`, inference is distributed across 6 Nvidia A40 GPUs, each with 48 GB of VRAM, totaling 288 GB. This distributed approach results in a total inference time of 126 GPU hours.

5.2 Results and discussion

Table 11 shows that `Google-BERT` slightly outperforms `Llama-3.2-3B` in the sense tagging accuracy (94.5% vs. 92.5 %). This suggests that the encoder-based architecture of BERT is well suited for the WSD task in Taiwan Mandarin. The annotation agreement between the two models, measured using Cohen's Kappa, is moderate at 0.664, indicating substantial but incomplete alignment in their outputs.

Not only does Google-BERT excel in performance, it also shows to be significantly more resource efficient. The model requires only a single GPU and completes inference in 17 hours, producing an estimated 1.69 kg CO_2eq. In comparison, the larger decoder-based model, Llama-3.2-3B, necessitates distributed inference across six GPUs to complete within a reasonable time frame, in a total computational cost of 126 GPU hours and an estimated 13.61 kg CO_2eq, which is approximately eight times the carbon footprint of Google-BERT. This distinction is crucial for deploying NLP technologies in an eco-friendly manner. In other words, `Google-BERT` achieves comparable results compared to `Llama-3.2-3B` with a significantly smaller resource footprint. This highlights a key trade-off between model size and computational cost. Although decoder-based models are powerful, their substantial resource demands may not be justifiable in tasks where encoder-based models achieve similar performance.

These findings contribute to ongoing discussions in the NLP community to prioritize sustainability in model development. A key consideration is the integration of energy consumption and computational cost metrics into existing evaluation frameworks. As emphasized in prior work (Strubell, Ganesh, and McCallum 2019; Castaño *et al.* 2023), selecting and designing models that

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

balance performance with environmental responsibility is essential. Recent automated tracking tools such as CodeCarbon (Schmidt *et al.* 2021) and Carbontracker (Anthony, Kanding, and Selvan 2020) have made per-experiment emissions monitoring increasingly accessible, lowering the barrier to systematic carbon reporting in NLP research. Practical measures, such as adopting lighter models and faster GPUs (Liu and Yin 2024), offer a promising way to reduce the carbon footprint of large-scale language models while maintaining robust model performance. Furthermore, establishing standardized frameworks for measuring and reporting the environmental impact of NLP systems, such as the Digital and Green Index (DGI) (Thelisson *et al.* 2023), is crucial for fostering transparency and accountability within the field. By incorporating metrics such as energy efficiency, carbon emissions, and GPU utilization into existing benchmarks, a more comprehensive evaluation of models becomes possible, considering both their technical capabilities and their environmental footprint. Such steps not only align with the growing demand for responsible AI development but contribute to broader sustainability goals, including the United Nations Sustainable Development Goal 13 (SDG 13) (Louman *et al.* 2019) on climate action. By integrating these considerations into the core of NLP research practices, the field can ensure that technological advancements are both impactful and environmentally conscious.

6. Conclusion

This study provides an assessment of encoder- and decoder-based models for WSD and RPD of proper nouns, using datasets from the CWN and two corpora in Taiwan Mandarin. Encoder-based models not only deliver comparable accuracy to decoders on WSD tasks, but they also exhibit greater computational efficiency. In contrast, decoder-based models, despite their higher resource requirements, perform worse than encoders on RPD tasks. The findings emphasize seeking balance among model performance, resource consumption, and environmental considerations. This highlights the importance of selecting models based on task-specific needs and resource constraints.

The implications of these findings extend beyond Taiwan Mandarin to other low-resource languages and multilingual NLP settings. Many languages lack the large annotated corpora and computational resources required to fine-tune and deploy large decoder-based models. Our results suggest that encoder-based models, which achieve comparable or superior performance at a fraction of the computational cost, represent a more practical and sustainable choice for such settings. This is particularly relevant for languages where lexical resources such as wordnets or sense inventories exist but training data remains scarce. By demonstrating that smaller, encoder-based models can match the performance of significantly larger decoders on fine-grained semantic tasks, this study provides a compelling case for prioritizing efficient architectures in the development of NLP tools for underrepresented languages. Furthermore, the substantially lower carbon footprint of encoder-based models aligns with the broader goal of democratizing NLP research, enabling institutions with limited resources, including those in developing regions, to conduct meaningful research without incurring prohibitive computational costs.

This study highlights the importance of considering both accuracy and resource consumption when selecting model architectures, particularly as large-scale models are increasingly being deployed. Future work should focus on optimizing models to achieve a balance between performance and efficiency, exploring techniques to minimize energy consumption while maintaining task accuracy. Integrating standardized sustainability metrics into evaluation frameworks is crucial to advancing environmentally conscious language models. The growing ecosystem of dedicated tracking tools offers a concrete path toward reproducible carbon reporting as a routine component of NLP benchmarking. This study demonstrates encoder-based models as being not only computationally efficient but also potentially less energy-intensive, advocating the use of smaller, task-focused, and sustainable models to reduce the environmental footprint of NLP systems.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Acknowledgments. The authors used AI language tools (ChatGPT, Gemini, and Claude) to assist with analysis scripts and light editorial revision of draft text. All scientific content, analysis, and conclusions are the authors' own.

References

Agirre E., **López de Lacalle O.** and **Soroa A.** (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* **40**(1), 57–84. https://doi.org/10.1162/COLI_a_00164. https://aclanthology.org/J14-1003

Anil R., **Dai A.M.**, **Firat O.**, **Johnson M.**, **Lepikhin D.**, **Passos A.**, **Shakeri S.**, **Taropa E.**, **Bailey P.**, **Chen Z.**, **et al.** (2023). Palm 2 technical report. arXiv preprint arXiv:2305.10403.

Anthony L.F.W., **Kanding B.** and **Selvan R.** (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. In ICML Workshop on Challenges in Deploying and Monitoring Machine Learning Systems. Available at https://arxiv.org/abs/2007.03051

Apresjan J.D. (1974). Regular polysemy. *Linguistics* **12**, 5–32. https://doi.org/10.1515/ling.1974.12.142.5.

Barba E., **Pasini T.** and **Navigli R.** (2021). ESC: Redesigning WSD with extractive sense comprehension. In Kristina T., Anna R., Luke Z., Dilek H.-T., Iz B., Steven B., Ryan C., Tanmoy C. and Yichao Z. (eds), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 4661–4672. https://doi.org/10.18653/v1/2021.naacl-main.371,

Blevins T. and **Zettlemoyer L.** (2020). Moving down the long tail of word sense disambiguation with gloss-informed biencoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 1006–1017. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.95

Boleda G., **Schulte im Walde S.** and **Badia T.** (2012). Modeling regular polysemy: A study on the semantic classification of Catalan adjectives. *Computational Linguistics* **38**(3), 575–616. https://doi.org/10.1162/COLI_a_00093

Bouza L., **Bugeau A.** and **Lannelongue L.** (2023). How to estimate carbon footprint when training deep learning models? A guide and review. *Environmental Research Communications* **5**(11), 115014. https://doi.org/10.1088/2515-7620/acf81b

Brown T.B., **Mann B.**, **Ryder N.**, **Subbiah M.**, **Kaplan J.**, **Dhariwal P.**, **Neelakantan A.**, **Shyam P.**, **Sastry G.**, **Askell A.**, et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, **33** (NeurIPS 2020). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Castaño J., **Martínez-Fernández S.**, **Franch X.** and **Bogner J.** (2023). Exploring the carbon footprint of hugging Face's ML models: A repository mining study. In 2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 1–12. https://doi.org/10.1109/ESEM56168.2023.10304801.

Chersoni E., **Lenci A.** and **Blache P.** (2017). Logical metonymy in a distributional model of sentence comprehension. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017). Vancouver, Canada: Association for Computational Linguistics, pp. 168–177. https://doi.org/10.18653/v1/S17-1021.

CKIP Lab (2020). Bert-base Chinese. Available at https://huggingface.co/ckiplab/bert-base-chinese (accessed 01 September 2025).

Conia S. and **Navigli R.** (2021). Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration. In Paola M., Jorg T. and Reut T. (eds), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Association for Computational Linguistics, pp. 3269–3275. https://doi.org/10.18653/v1/2021.eacl-main.286.

Del Tredici M. and **Bel N.** (2015). A word-embedding-based sense index for regular polysemy representation. In Phil B., Shay C., Paramveer D. and Percy L. (eds), *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Denver, Colorado: Association for Computational Linguistics, pp. 70–78. https://doi.org/10.3115/v1/W15-1510.

Devlin J., **Chang M.-W.**, **Lee K.** and **Toutanova K.** (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill B., Christy D. and Thamar S. (eds), *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)*, Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Energy Administration, Ministry of Economic Affairs (2024). Energy statistics handbook 2024 能源統計手冊. Technical report, Taipei, Taiwan: Energy Administration (能源署). Available at https://ea01.moeaea.gov.tw/a0303/02/attachments/handbook/2024/ebook/2024EnergyStaHandBook.pdf.

Gemma Team, **Mesnard T.**, **Hardin C.**, **Dadashi R.**, **Bhupatiraju S.**, **Pathak S.**, **Sifre L.**, **Rivière M.**, **Kale M.S.**, **Love J.**, **et al.** (2024). Gemma: Open models based on Gemini research and technology. arXiv preprint arXiv:2403.08295.

Google Research (2018). Bert-base Chinese. Available at https://huggingface.co/google-bert/bert-base-chinese (accessed 01 September 2025).

He P., **Gao J.** and **Chen W.** (2023). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations (ICLR 2023),* Kigali, Rwanda, May 1–5. OpenReview.net. https://openreview.net/forum?id=sE7-XhLxHA.

Hosseini M., **Hosseini M.** and **Javidan R.** (2024). Leveraging large language models for clinical abbreviation disambiguation. *Journal of Medical Systems* **48**(1), 27.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Hsieh S.-K., **Tseng Y.-H.**, **Chou H.-Y.**, **Yang C.-W.** and **Chang Y.-Y.** (2024). Resolving regular polysemy in named entities. arXiv preprint arXiv:2401.09758

Hu E.J., **Shen Y.**, **Wallis P.**, **Allen-Zhu Z.**, **Li Y.**, **Wang S.**, **Wang L.** and **Chen W.** (2022). LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations. https://openreview.net/forum?id=nZeVKeeFYf9

Huang L., **Sun C.**, **Qiu X.** and **Huang X.** (2019). GlossBERT: BERT for word sense disambiguation with gloss knowledge. In Kentaro I., Jing J., Vincent N. and Xiaojun W. (eds), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3509–3514. https://doi.org/10.18653/v1/D19-1355.

Kang H., **Blevins T.** and **Zettlemoyer L.** (2024). Translate to disambiguate: Zero-shot multilingual word sense disambiguation with pretrained language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024)*, Volume 1: Long Papers. Association for Computational Linguistics. https://aclanthology.org/2024.eacl-long.176.

Kugic A., **Schulz S.** and **Kreuzthaler M.** (2024). Disambiguation of acronyms in clinical narratives with large language models. *Journal of the American Medical Informatics Association* **31**(9), 2040–2046. https://doi.org/10.1093/jamia/ocae157.

Laurer M., **van Atteveldt W.**, **Casas A.** and **Welbers K.** (2024). Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and BERT-NLI. *Political Analysis* **32**(1), 84–100. https://doi.org/10.1017/pan.2023.20.

Lin Y.-T. (2023). Bert-base traditional Chinese. Available at https://huggingface.co/yentinglin/bert-base-zhtw (accessed 01 September 2025).

Liu V. and **Yin Y.** (2024). Green AI: Exploring carbon footprints, mitigation strategies, and trade offs in large language model training. *Discover Artificial Intelligence* **4**, 49. https://doi.org/10.1007/s44163-024-00149-w.

Liu Y., **Ott M.**, **Goyal N.**, **Du J.**, **Joshi M.**, **Chen D.**, **Levy O.**, **Lewis M.**, **Zettlemoyer L.** and **Stoyanov V.** (2019). RoBERTa: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Louman B., **Keenan R.J.**, **Kleinschmit D.**, **Atmadja S.**, **Sitoe A.A.**, **Nhantumbo I.**, **de Camino Velozo R.** and **Morales J.P.** (2019). SDG 13: Climate action – impacts on forests and people. In Pia K., Carol J.P.C., Wil de J., Glenn G., Pablo P. and Georg W. (eds), *Sustainable Development Goals: Their Impacts on Forests and People*. Cambridge University Press, pp. 419–444.

Luccioni A.S., **Viguier S.** and **Ligozat A.-L.** (2023). Estimating the carbon footprint of BLOOM, a 176B parameter language model. *Journal of Machine Learning Research* **24**(253), 1–15. http://jmlr.org/papers/v24/23-0069.html

Meta (2024). Llama 3.2 3b. Available at https://huggingface.co/meta-llama/Llama-3.2-3B (accessed 01 September 2025).

Open A.I., **Achiam J.**, **Adler S.**, **Agarwal S.**, **Ahmad L.**, **Akkaya I.**, **Aleman F.L.**, **Almeida D.**, **Altenschmidt J.**, **Altman S.**, **et al.** (2024). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Patterson D., **Gonzalez J.**, **Holzle U.**, **Le Q.**, **Liang C.**, **Munguia L.-M.**, **Rothchild D.**, **So D.R.**, **Texier M.** and **Dean J.** (2022). The carbon footprint of machine learning training will plateau, then shrink. *Computer* **55**(7), 18–28. https://doi.org/10.1109/MC.2022.3148714.

Peters M.E., **Neumann M.**, **Iyyer M.**, **Gardner M.**, **Clark C.**, **Lee K.** and **Zettlemoyer L.** (2018). Deep contextualized word representations. In Marilyn W., Heng J. and Amanda S. (eds), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. https://doi.org/10.18653/v1/N18-1202,

Pustejovsky J. (1995). *The generative Lexicon*. The MIT Press. https://doi.org/10.7551/mitpress/3225.001.0001.

Rajpoot P., **Jindal A.** and **Parikh A.** (2024). Upaya at ArabicNLU shared-task: Arabic lexical disambiguation using large language models. In Nizar H., Houda B., Ramy E., Nadi T., Ibrahim A.F., Ahmed A., Samia T., Injy H., Yaser O., Bashar A., Wissam A., Salam K., Hatem H., Imed Z., Badr A.K., Rawan A. and Khalil M. (eds), *Proceedings of The Second Arabic Natural Language Processing Conference*, Bangkok, Thailand: Association for Computational Linguistics, pp. 377–382. https://doi.org/10.18653/v1/2024.arabicnlp-1.32

Schmidt, **V.**, **Goyal**, **K.**, **Joshi**, **A.**, **Feld**, **B.**, **Conell**, **L.**, **Laskaris**, **N.**, **Blank**, **D.**, **Wilson**, **J.**, **Friedler**, **S.**, & **Luccioni**, **S.** (2021). CodeCarbon: Estimate and track carbon emissions from machine learning computing. arXiv preprint arXiv:2102.06670.

Scozzafava F., **Maru M.**, **Brignone F.**, **Torrisi G.** and **Navigli R.** (2020). Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In **Asli C.** and **Tsung-Hsien W.** (eds), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics, pp. 37–46. https://doi.org/10.18653/v1/2020.acl-demos.6,

Shi Y., **Shu P.**, **Liu Z.**, **Wu Z.**, **Li Q.**, **Liu T.**, **Liu N.** and **Li X.** (2024). MGH radiology llama: A llama 3 70b model for radiology. arXiv preprint arXiv:2408.11848.

Strubell E., **Ganesh A.** and **McCallum A.** (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pp. 3645–3650, Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1355.

Thelisson E., **Mika G.**, **Schneiter Q.**, **Padh K.** and **Verma H.** (2023). Toward responsible AI use: Considerations for sustainability impact assessment. arXiv preprint arXiv:2312.11996.

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Wang M. and **Wang Y.** (2020). A synset relation-enhanced framework with a try-again mechanism for word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6229–6240.

Wang M. and **Wang Y.** (2021). Word sense disambiguation: Towards interactive context exploitation from both word and sense perspectives. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, pp. 5218–5229. https://doi.org/10.18653/v1/2021.acl-long.406,

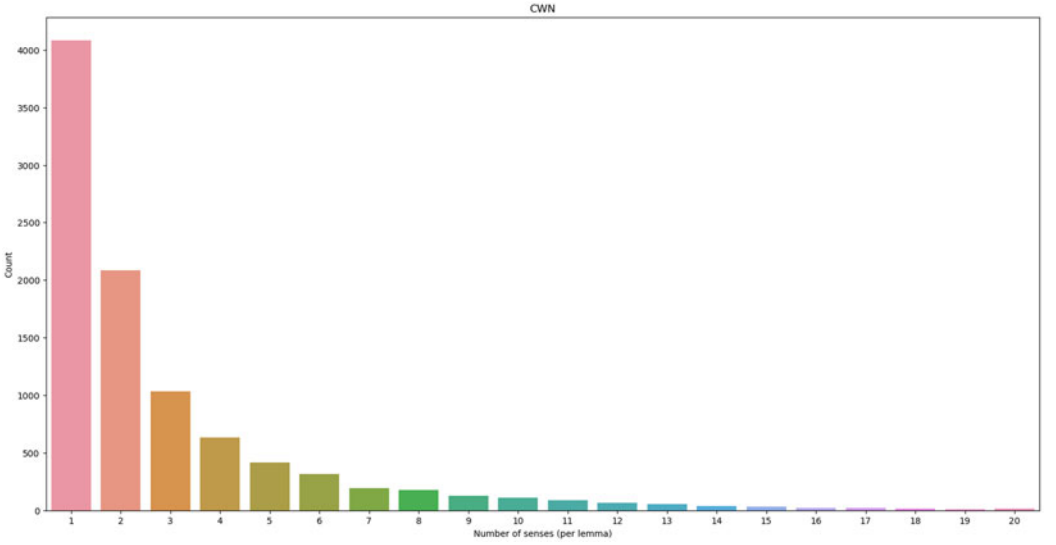
Wolf T., **Debut L.**, **Sanh V.**, **Chaumond J.**, **Delangue C.**, **Moi A.**, **Cistac P.**, **Ma C.**, **Jernite Y.**, **Plu J.**, **Xu C.**, **Scao T.L.**, **Gugger S.**, **Drame M.**, **Lhoest Q.** and **Rush A.M.** (October 2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics*, pp. 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

Yae J., **Skelly N.**, **Ranly N.** and **LaCasse P.** (2024). Leveraging large language models for word sense disambiguation. *Neural Computing and Applications* **12**, 1–18. https://doi.org/10.1007/s00521-024-10747-5.

Zhang, **J.**, **Gan**, **R.**, **Wang**, **J.**, **Zhang**, **Y.**, **Zhang**, **L.**, **Yang**, **P.**, **Gao**, **X.**, **Wu**, **Z.**, **Dong**, **X.** et al. (2022). Fengshenbang 1.0: Being the foundation of Chinese cognitive intelligence. arXiv preprint arXiv:2209.02970.

Zhang M. (Mxode) (2023). Smollm-Chinese-180m. Available at https://huggingface.co/Mxode/SmolLM-Chinese-180M (accessed 01 September 2025).

Appendix A. Sense distribution in CWN



Appendix B. Decoder-based model input structure

Table B1. The special tokens used for decoder-based models. `BOS` is used at the beginning of an example and to separate the first and second sequences. `CLS` indicates the end of the example. The model uses `CLS` as input for the classifier head

Model	BOS	CLS
SmolLM-180M	<|start_of_text|>	<|unused159|>
Gemma-2-2b	<bos>	<unused99>
Llama-3.2-3B	<|begin_of_text|>	<|eot_id|>

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Appendix C. Models used

Table C1. All models were trained on one GPU (24 GB VRAM) either locally or through a cloud GPU provider. `Type` refers to if the model is an encoder (**E**) or a decoder (**D**) model.

Hugging face hub repository (simplified name)	Type	Params	GPU	Training time
ckiplab/bert-base-chinese (CkipLab-BERT; CKIP Lab, 2020)	E	103M	RTX 4090	50m
google-bert/bert-base-chinese (Google-BERT; Google Research, 2018)	E	103M	RTX 4090	50m
IDEA-CCNL/Erlangshen-DeBERTa-v2-97M-Chinese (Erlangshen-DBT-v2; Zhang *et al*. 2022),	E	97.7M	RTX 4090	2h 13m
microsoft/deberta-v3-base (He et al. 2023)	E	184M	RTX 4090	2h 9m
microsoft/deberta-v3-small (He *et al.* 2023)	E	142M	RTX 4090	1h 12m
microsoft/deberta-v3-large (DBT-v3-large; He *et al.* 2023)	E	435M	RTX 4090	5h 5m
microsoft/mdeberta-v3-base (He *et al.* 2023)	E	276M	RTX 4090	1h 44m
MoritzLaurer/mDeBERTa-v3-base-mnli-xnli (Laurer *et al.* 2024)	E	279M	RTX A5000	3h 3m
MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 (mDBT-v3-xnli; Laurer *et al.* 2024)	E	279M	RTX 4090	1h 46m
yentinglin/bert-base-zhtw (yentinglin-BERT; Lin, 2023)	E	110M	RTX 4090	50m
gogle/Gemma-2-2b† (Gemma Team, 2024)	D	2.61B	RTX A5000	19h 16m
meta-llama/Llama-3.2-3B† (Meta, 2024)	D	3.21B	RTX 4090	12h 48m
Mxode/SmolLM-Chinese-180 M (SmolLM-180M; Zhang 2023)	D	177M	RTX A5000	6h 19m

† indicates the model was trained using LoRA. This table contains all models that were trained. The results presented in the main text only present the model with the best performance within a family of models (e.g., only Microsoft's `DBT-v3-large` is present when there are actually `small`, `base`, and `large` variants

Appendix D. Hyperparameters

Table D1. The hyperparameters used to train the *majority* models with Transformers' `Trainer`. Note: `Mxode/SmolLM-Chinese-180M` was trained in FP32 because of training instability when using BF16

Hyperparameter	Value
Epochs	3
Learning rate	5e-5
LR scheduler	cosine
Warm-up ratio	0.1
Optimizer	adamw_8bit
Precision	bfloat16
Weight decay	0.1

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press

Table D2. The LoRA hyperparameters used to train `Llama-3.2-3B` and `google-Gemma-2-2b`

LoRA hyperparameter	Value
LoRA rank	8
LoRA alpha	16
LoRA dropout	0.1
target modules	all-linear
rsLoRA	True
Init LoRA weights	Gaussian
load in 4-bit	True
4-bit quant type	nf4
Use double quant	True
4-bit compute dtype	bfloat16

Cite this article: Chen P-E, Lian D-C and Hsieh S-K (2026). Balancing accuracy and efficiency: Evaluating encoder- and decoder-based models for word sense disambiguation and regular polysemy detection. *Natural Language Processing* **32**, 451–469. https://doi.org/10.1017/nlp.2026.10033

https://doi.org/10.1017/nlp.2026.10033 Published online by Cambridge University Press