Expanding WordNet Based on Glosses: Methodology and Applications

Yicheng Sun and Jie Wang
Richard Miner School of Computer and Information Sciences
University of Massachusetts, Lowell, MA 01854, USA
Correspondence: Jie_Wang@uml.edu

Abstract

We introduce a method called AI-WordNet for expanding WordNet using only glosses. Given a lemma and its gloss, AI-WordNet predicts the corresponding lexname for the gloss, decides whether a new synset should be created for the lemma, finds the best hypernym for the new synset, and updates the corresponding hypernym-hyponym relationships. We demonstrate a low-cost implementation of AI-WordNet for noun lemmas using PWN 3.0 as the training dataset. Intrinsic evaluations show that AI-WordNet achieves a high F1 score of 94.6% for lexname prediction, and 64.8% direct hits on true hypernyms with an average distance of 1.374 from the predicted hypernyms to the true hypernyms. We apply AI-WordNet to generate distractors for cloze-question creation on answer keys containing new lemmas not yet included in PWN 3.0, using glosses extracted from Wiktionary. Extrinsic evaluations confirm the high quality of the generated distractors.

1 Introduction

Using WordNet to complete an NLP task offers the advantage of achieving a controllable and explainable solution. For instance, WordNet has been used to automate the generation of distractors for constructing cloze questions based on an answer key derived from a given text. Cloze questions are fill-in-the-blank exercises where each blank is accompanied by multiple options, including distractors—incorrect but plausible choices—alongside the correct option known as the answer key. In this approach, distractors are constructed from lemmas in WordNet that are semantically related to the lemmas in the answer key within a specified distance. This ensures a clear rationale for their selection by leveraging the semantic relationships between the lemmas in the answer key and those in the distractors.

However, not all lemmas of interest are included in WordNet. Therefore, an effective and low-cost solution for inserting new lemmas into WordNet is highly desirable, ensuring that the existing system for completing the NLP task can continue to function seamlessly.

Adding a new lemma to WordNet requires information about its gloss, as well as its gloss's lexname, synset, hypernyms, and hyponyms. While this can be achieved by curating authoritative digitized dictionaries or datasets with predefined lexical relations, emerging lemmas are often not yet included in these sources. Even when they are present, they are typically represented as standalone entries without systematic semantic connections or with relationships tied to broader lemmas rather than being specifically linked to their glosses.

Addressing these issues, we present a method called AI-WordNet to facilitate the insertion of new lemmas into WordNet using only their glosses. These glosses can be easily sourced from Wiktionary or other publicly available databases, enabling a streamlined and efficient approach to expanding WordNet's lexical coverage.

AI-WordNet performs the following tasks: It first predicts a lexname for the gloss. If the lemma is new but the gloss already exists in WordNet, it joins the lemma to the appropriate synset. Otherwise, it creates a new synset for the lemma. To find a hypernym synset for the newly created synset, it predicts a path of inherited hypernyms. It then checks whether any of these predicted hypernyms are already in WordNet. If a match is found, it selects the best-matched hypernym synset. Otherwise, it searches for the closest matching hypernym gloss among existing glosses and maps it to its corresponding hypernym synset. Finally, It updates the relevant hypernym-hyponym relationships to incorporate the newly created synset.

We demonstrate a low-cost implementation of AI-WordNet for noun lemmas using various open-source pre-trained small models (PLMs) fine-tuned on datasets constructed from PWN 3.0 (Princeton WordNet v3.0), which can be run on a commonplace GPU server.

We show through intrinsic evaluations that AI-WordNet achieves a high F1 score of 94.6% for lexname prediction, and 64.8% direct hits on the true hypernyms with the average distance of 1.374 from the predicted hypernyms to the true hypernyms. Additionally, we apply AI-WordNet to generate controlled and explainable distractors using the cloze-question-generation system (Sun and Wang, 2023) on texts containing new lemmas with glosses extracted from Wiktionary. Extrinsic evaluations confirm the high quality of the generated distractors, as errors in lexname predictions and hypernym identifications, as long as they remain within a reasonable range of the ground truth, are acceptable for distractor generation without compromising overall effectiveness.

2 Related Work

Sun and Wang (Sun and Wang, 2023) presented a system called Cloze Question Generator (CQG) for constructing cloze questions from a given article, with a particular emphasis on generating multigram distractors using the semantic structure of WordNet. This approach allows users to control how distractors are generated and provides explanations for the appropriateness of the generated distractors based on the WordNet structure.

In particular, given an answer key for a stem (a sentence with blanks to fill in), CQG first segments the answer key into a sequence of instances. For each instance, it generates instance-level distractor candidates using a transformer and sibling synsets in WordNet, and ranks them based on contextual similarities, synset relations, and lexical relatedness. Distractor candidates are then formed by selectively replacing instances with the top-ranked instance-level candidates, which are subsequently checked for legitimacy as phrases. Finally, CQG selects the top-ranked distractor candidates as distractors based on contextual semantic similarities to the answer key. This process is controllable, and the selection of distractors can be explained based on the WordNet structure, specifically by examining how semantically distant the distractors are from the answer key.

Intrinsic evaluations demonstrated that CQG significantly outperforms previous methods, and extrinsic evaluations also confirmed the high quality of the generated distractors.

To the best of our knowledge, no published work has reported automatic insertions of new lemmas into WordNet using glosses as the sole source of information, although the white papers of both BabelNet (Navigli and Ponzetto, 2012) and ConceptNet (Speer et al., 2016) indicate that new entries can be added automatically. However, the methods employed remain undisclosed. On the other hand, Koeva (Svetla, 2021) demonstrated how to expand WordNet using conceptual frames.

On a separate note, cloze questions can be generated from a given text using pre-trained general-purpose large language models (LLMs), such as GPT-4. However, this approach functions as a black box, offering limited control over the generation of distractors and minimal explainability regarding the rationale behind their selection. Furthermore, when answer keys contain lemmas absent from the training dataset of the underlying LLM, the model is unable to generate suitable distractors. Resolving this issue often requires retraining the model, a process that is both astronomically expensive and time-consuming.

3 AI-WordNet Overview

For emerging terms or phrases, such as new Internet slang or academic terminology that have not yet been included in any authoritative dictionary, we classify them as lemmas.

In what follows, unless otherwise stated, direct hypernyms will be referred to simply as hypernyms. Direct hypernyms, as well as any hypernyms beyond the direct hypernym level—such as hypernyms of hypernyms—are collectively referred to as inherited hypernyms.

AI-WordNet consists of six components (Fig. 1):

(1) The **Lexname Predictor** predicts a lexname L for the gloss g.

(2) The **Synset Identifier** identifies an appropriate synset for (l, g). If l is in WordNet and there is an existing synset for l with a gloss that has the same meaning as g, then it drops (l, g). Otherwise, it checks if there is a synset S for which $l \notin S$, but with a gloss that shares the same meaning as g. If so, place l in S. If not, create a new synset $S_{l,g}$.

Note that the lexname predicted for g by the

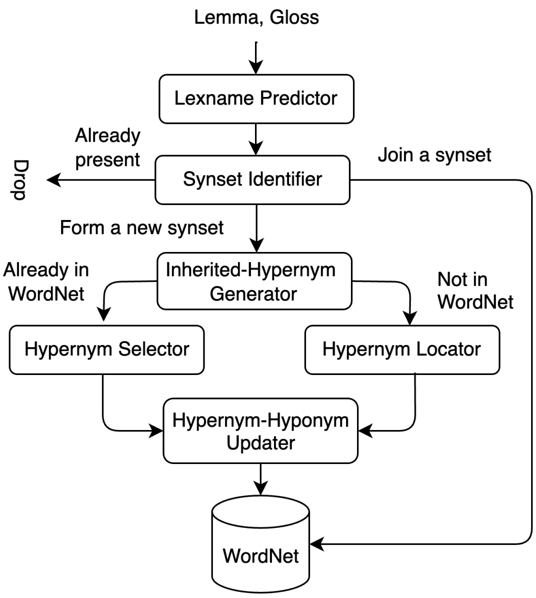

Figure 1: Schematic of AI-WordNet and data flow

Lexname Generator reduces the search space to only include synsets with the same predicted lexname for (l, g). This is the reason for the Lexname Predictor to precede the Synset Identifier.

(3) The **Inherited-Hypernym Generator** generates a hypernym h for the new synset $S_{l,g}$. Depending on whether h is in WordNet, the system goes to the Hypernym Selector or the Hypernym Locator.

(4) The **Hypernym Selector** determines, for the given h, an appropriate synset from (possible) multiple synsets of h contained in WordNet.

(5) The **Hypernym Locator** locates the best a synset in WordNet as g's hypernym.

(6) The **Hypernym-Hyponym Updater** updates the hypernym-hyponym structure of WordNet when new synsets are inserted.

4 Implementation

Nouns and phrasal nouns are the most relevant as answer keys for constructing cloze questions. We present a low-cost implementation of AI-WordNet for noun lemmas using small open-source PLMs that can be fine-tuned.

Implementations for lemmas of other parts of speech, such as verbs and phrasal verbs, can be carried out using the same approach.

4.1 Fine-tuning Through OpenPrompt

We fine-tune PLMs directly for various generation tasks and indirectly for various classification tasks using the OpenPrompt (OP) platform (Ding et al., 2021). OP uses the mechanism of Verbalizer to enhance fine-tuning.

Let M be a PLM supported by OP. Denote by OP/M a fine-tuned model of M through OP over a dataset and Verbalizer with the following general prompt template (PT):

> [text] {'place_holder': text_1}, ..., [text] {'place_holder': text_n}. [text] [mask] [text].

Here {'place_holder': text_k} represents a variable and [text] a text segment without variables, which could be empty. A Verbalizer includes terms positively or negatively associated with the target to be predicted.

On a training dataset, OP automatically replaces the variables with data entries in the dataset one at a time, and uses M to predict what is in the place of the [mask]. The predicted term, if not in the Verbalizer, is first mapped to a term in the Verbalizer through a neural network and then mapped to a target term through another neural layer.

In what follows, we choose T5 as the default PLM due to its strong performance and its availability as a no-cost, open-source model.

4.2 Datasets

The primary dataset, denoted by WordNet-N, consists of all noun synsets in PWN 3.0 together with lexnames, glosses, hypernym synsets, and hyponym synsets. It contains a total of 82,061 data entries, which are organized into 26 classes corresponding to the 26 lexnames for noun synsets.

We will later construct various datasets from WordNet-N, which maintain the same proportion across classes to ensure consistent class distribution. Each dataset will be named using the abbreviation of its corresponding component followed by "DS." For instance, the dataset for training and testing the Lexname Predictor will be named LP-DS. These datasets are published on github [1]. Each dataset will be split following the standard 80-20 ratio for training and testing.

4.3 Lexname Predictor (LP)

The LP-DS dataset consists of all gloss-lexname pairs extracted from WordNet-N, except the lexname 'noun.Tops', which represents the highest category for any noun glosses.

Fine-tuning T5 and OP/T5 uses, respectively, the following PTs:

[1] https://github.com/wordneter/dataset

> What is the lexname of the given gloss?
> GLOSS: {gloss}.
> {'place_holder': gloss}. The lexname for the gloss is [mask].

The Verbalizer specified for each of the 25 lexnames consists of just the suffix. For example: the Verbalizer for lexname 'noun.person' is {'person'}.

LP first predicts a lexname L for g via the fine-tuned T5. If L is a suffix of a noun lexname, use 'noun.L' as the lexname for g. Otherwise, use the fine-tuned OP/T5 to predict a lexname for g.

4.4 Synset Identifier (SI)

The SI-DS dataset is constructed as follows: For each gloss in WordNet-N, identify the corresponding gloss with the same meaning from a digitized Oxford Dictionary. Use the low-cost GPT-3.5-Turbo to generate glosses with different wordings that express the same meaning, and glosses that express different meanings with various degrees of differences. Select independently at random 10,000 pairs of glosses that convey the same connotations (meaning) and label them as 1. Similarly, select independently at random 10,000 pairs of glosses that have different connotations and label them as 0.

SI is an OP/T5 binary classifier fine-tuned on SI-DS using the following PT:

> Sentence 1: {gloss 1}.
> Sentence 2: {gloss 2}.
> The meanings of sentence 1 and sentence 2 are [mask].

For the label 1, the Verbalizer includes {'synonymous', 'equivalent', 'identical', 'interchangeable', 'coincident', 'matching'}. For the label 0, it includes {'disparate', 'divergent', 'distinct', 'dissimilar', 'unalike', 'incompatible', 'varied'}. Note that the word 'alike' is not precise enough for label 1, whereas 'unalike' is acceptable for label 0.

Note that we might be tempted to directly use cosine similarity of sentence embeddings for glosses to identify whether g is similar to an existing gloss. However, different glosses can have a cosine similarity score close to 1 under BERT embeddings as the following example shows: The glosses 'United States actor; son of Maurice Barrymore and Georgiana Barrymore (1878-1954)' and 'United States actor; son of Maurice Barrymore and Georgiana Barrymore (1882-1942)' for Synset(barrymore.n.01) and Synset(barrymore.n.03) have a cosine similarity of 0.999 under BERT embeddings. Setting the threshold this high would result in incorrectly identifying certain glosses with the same meaning as different, rendering this approach unsuitable.

4.5 Inherited-Hypernym Generator (IHG)

The IHG-DS dataset is constructed as follows: Initially, for each synset in WordNet-N, use its gloss as the source and the path of its inherited hypernyms as the target, where the path starts from the direct hypernym of the synset.

Glosses can be categorized into two types: those containing inherited hypernyms and those that do not. Specifically, 83.4% of glosses in WordNet-N fall into the first category, with 73.5% of them containing direct hypernyms. This imbalance between the two types causes a model trained on the initial dataset to favor extracting nouns or phrasal nouns from glosses as hypernyms, negatively impacting the generation of hypernyms for glosses in the second category, as none of the extracted terms is an appropriate hypernym.

To achieve a better balance, we use GPT-3.5-Turbo to generate, for each gloss, three differently articulated versions, and add these generated glosses to the training dataset if they contain no inherited hypernyms. This improves the ratio of the first and the second categories from 87:13 to 63:37. Note that this data augmentation is performed exclusively on the training dataset.

Note that all paths of inherited hypernyms for noun lemmas eventually converge at the same root node, 'entity', which has the lexname 'noun.Tops'. As a result, a model trained on the initial dataset tends to extract a lemma from the path closer to the root as the hypernym. To mitigate this issue, we extract the initial k nodes of the hypernym path as the target text, and we refer to it as a k-gram.

Fine-tuning OP/T5 uses the following PT on the augmented training dataset:

> Generate {k} inherited hypernyms for a given gloss. Sort them based on their proximity to the gloss in terms of meaning, separated by symbol →. GLOSS: {gloss}.

The first lemma in the k-gram is designated as the *predicted hypernym* of the underlying gloss. For example, for the lemma-gloss pair ('apple', 'fruit with red or yellow or green skin and sweet to tart crisp whitish flesh'), IHG outputs the following path:

'edible fruit → fruit → produce',

where 'edible fruit' is the predicted hypernym.

We found that the predicted hypernym may vary depending on the values of k. We select $k = 3$ based on the experiments we conducted with various values of k, ranging from 1 to the full path length, which indicates that $k = 3$ maximizes the probability of the predicted hypernym being the true hypernym.

Let h be the predicted hypernym in the 3-gram. If it is in WordNet-N, then pass it to the Hypernym Selector. Otherwise, branch to the Hypernym Locator.

4.6 Hypernym Selector (HS)

Let L be the predicted lexname by LP, $S_{l,g}$ be either a new synset or an existing synset identified by SI, and h be what is passed by IHG. If h appears in only one synset, then this synset is the hypernym synset of $S_{l,g}$. If h appears in multiple synsets $S_{h,1}, \ldots, S_{h,m}$ with $m > 1$ and $S_{h,i}$ having gloss g_i, we use HS to select an appropriate synset S_{h,g_i} as the hypernym synset of $S_{l,h}$.

HS is a fine-tuned ESCHER (Barba et al., 2021) model on the HS-DS dataset, where ESCHER is a transformer-based model for extractive sense comprehension with superior performance over similar word-sense-disambiguation (WSD) tools.

Each data entry in HS-DS consists of three components: input text, target location, and ground truth. The input text is a sentence that combines the hypernym and the target gloss using the following template:

> [hypernym] is a hypernym of the gloss: [target gloss].

The template provides the model with the necessary context for understanding the relationship between the gloss and its hypernyms. The target location specifies the location of the hypernym within the input text for the model to identify and focus on the relevant part of the input text during fine-tuning. The ground truth is the actual synset name of the hypernym of the underlying gloss.

HS takes h, g, and g_i's as input and returns the gloss that is the closest to the hypernym of g. For example, let g be 'Fruit with red or yellow or green skin and sweet to tart crisp whitish flesh with h being 'fruit'. The lemma 'fruit' appears in three synsets with glosses being, respectively, 'The ripened reproductive body of a seed plant', 'An amount of a product', and 'The consequence of some effort or action'. The gloss 'The ripened reproductive body of a seed plant' is the closest to g determined by HS, which is returned as the output.

4.7 Hypernym Locator (HL)

HL is branched if the predicted hypernym is not in WordNet-N. In this case, it is still possible that the true hypernym of the gloss g, denoted by h, exists in WordNet-N. This would mean that there is another gloss g' in WordNet-N with h being its hypernym, indicating that h is not a leaf node. Thus, g' can be identified by traversing all glosses in WordNet-N and applying an embedding-based method, such as BERT embeddings, to find the gloss most similar to g based on cosine similarity. The hypernym of g' would then be designated as the hypernym of g.

Unfortunately, h could also be a leaf node in WordNet-N, then g' is likely h itself, as g' may likely be the most similar to g. In such a scenario, running the above algorithm assigns h's hypernym h' as g's direct hypernym (see Fig. 2). This should be avoided to preserve the correct hierarchical structure.

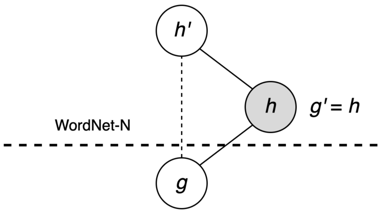

Figure 2: Assigning h' as the direct hypernym to g should be avoided, where the shaded node h is a leaf node in WordNet-N

To address this issue, HL uses LLM-Embedder (Zhang et al., 2023) fine-tuned on the HL-DS dataset. LLM-Embedder is an embedding method with a specifically pre-trained LLM through contrastive learning and knowledge distillation.

Each data point in the HL-DS dataset consists of (1) a gloss in WordNet-N; (2) an instruction to find the best hypernym gloss for the given gloss; (3) the synset for the true hypernym gloss of the given gloss called the query synset; (4) a set of glosses that are not direct hypernyms of the query synset; and (5) a set of glosses connected to the query synset with a distance at most 2, excluding direct hypernym glosses.

HL, on input g, searches from the set of all glosses in WordNet-N that have the same lexname as that of g for the best hypernym gloss of g, using the same instruction as in the training dataset with g being the gloss for the query synset. In particular, HL first combines the instruction and the query to create an augmented query, which is transformed into an embedding vector by LLM-Embedder. It generates concurrently embedding vectors for each gloss in the underlying set of glosses. Next, HL computes the cosine similarity between the embedding vector of the augmented query and the embedding vector of each gloss in the gloss set. The gloss with the highest similarity score is deemed the best hypernym gloss for the input gloss g. HL outputs the synset of this gloss.

4.8 Hypernym-Hyponym Updater (HHU)

After a new synset $S_{l,g}$ is inserted into WordNet-N, if an original synset $S_{l',g'}$ becomes a sibling of $S_{l,g}$, and both glosses g and g' share a common lemma, yet $S_{l,g}$ did not select $S_{l',g'}$ as its hypernym, then it would mean that $S_{l,g}$ should become the direct hypernym of $S_{l',g'}$. This situation can be illustrated using the following example:

Suppose that the following new lemma l with the gloss g is added to WordNet, with 'platform' identified as its direct hypernym, where

> $l =$ 'social media platform'.
>
> $g =$ 'an online digital space that allows individuals and groups to connect, interact, and share content with each other. These platforms enable users to create profiles, share text, photos, videos, and other multimedia forms, and partake in various communication and networking modes'.

Now, assume that prior to this insertion, the lemma $l' =$ 'TikTok' with the gloss $g' =$ 'a social media platform that allows users to create, share, and discover short-form videos' was already inserted, and 'platform' was identified as its direct hypernym. Now $S_{l',g'}$ becomes a sibling of $S_{l,g}$. Since g and g' share the same lemma 'platform', the fact that $S_{l,g}$ did not select $S_{l',g'}$ as its hypernym implies that $S_{l,g}$ (for 'social media platform') is a more specific hypernym of $S_{l',g'}$ (for 'TikTok'). Hence, it is necessary to perform this update to ensure that the hypernym-hyponym relationships are properly restored, maintaining consistency in the semantic network (see Fig. 3).

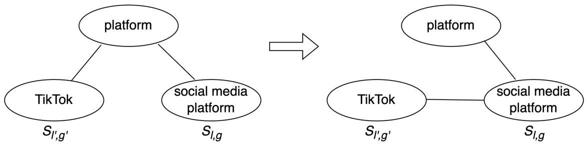

Figure 3: An illustration of updating hypernyms

HHU rectifies the hypernym-hyponym associations as follows: After a new synset $S_{l,g}$ is inserted, check for each sibling synset $S_{l',g'}$ whether g and g share a common lemma. If yes, then set $S_{l,g}$ to be the direct hypernym of $S_{l',g'}$. Otherwise, no changes are made.

5 Evaluations

AI-WordNet takes about 3 seconds in the worst case to insert a new lemma into WordNet with a given gloss on a machine equipped with an A6000 GPU and an i9 CPU. We evaluate the accuracy of AI-WordNet through both intrinsic and extrinsic evaluations.

5.1 Intrinsic Evaluations

Intrinsic evaluations use the standard metric of **F1 score** to assess the performance of the classification task in predicting lexnames. For the generation task in predicting hypernyms, two metrics are used: **direct hits** on the ground truth, which measures exact matches; and **distance** from the ground truth, which accounts for how far the predicted hypernyms deviate from the correct ones.

Specifically, the metric of direct hits is the percentage of predicted hypernyms that exactly match the true hypernyms of glosses, denoted by $H(1, 1)$. The metric of distance is the length of the shortest path between the predicted hypernym and the true hypernym of the underlying gloss in the test set. Thus, a distance of 0 means a direct hit, the distance of 1 means the predicted hypernym is a hypernym of the true hypernym, and a distance of 2 means the predicted hypernym is either a sibling or the grand-hypernym of the true hypernym (see Fig. 4).

Let H-AvgD denote the average distance between the predicted hypernyms and the true hypernyms, with smaller values indicating better performance.

Table 1: Intrinsic evaluations of AI-WordNet

Lexname F1	$\boldsymbol{H(1,1)}$	**H-AvgD**
0.946	0.648	1.374

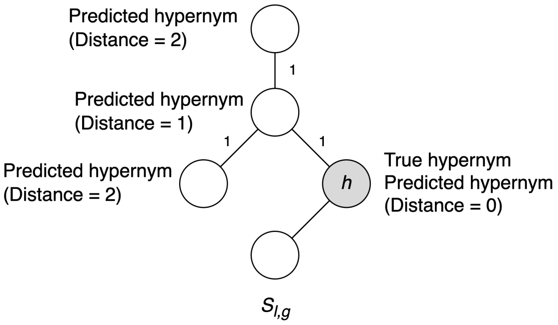

Figure 4: Distance illustration; the shaded node is the true hypernym

It can be seen from Table 1 that the low-cost implementation of AI-WordNet achieves a high F1 score on lexname predictions, and 64.5% direct hits with an average distance of 1.374, indicating that the predicted hypernyms are closely related to the corresponding true hypernyms.

5.2 Extrinsic Evaluations

Observe that the distance falls within a small range. These minor discrepancies would unlikely negatively impact the generation of reasonable distractors (Section 5.3 provides more information).

We randomly select 8,000 noun entries from Wiktionary at https://www.wiktionary.org/ that are not included in WordNet-N, and insert them using AI-WordNet. Each entry consists of a lemma-gloss pair, where the gloss is the definition provided in Wiktionary. Instead of manually verifying whether these entries are correctly integrated into WordNet, a process that demands advanced linguistic and domain expertise beyond our scope, we assess the contextual appropriateness of the distractors generated for the new lemmas within the answer keys of the constructed cloze questions.

Specifically, we select independently at random 300 of the constructed cloze questions using the CQG system (Sun and Wang, 2023), where the stem sentences for the cloze questions are extracted from the history section of Wiktionary, which includes the corresponding lemmas.

Three human judges were recruited to evaluate the reliability and plausibility of distractors generated by CQG using the following rubric. They have all written exam questions with at least three years of teaching experience at the college level.

• **Reliability**: A distractor receives a score of 1 if placing it in the blank space of the stem results in a contextually appropriate and grammatically correct, yet logically incorrect, sentence. If it fails to meet these criteria, it receives a score of 0.

• **Plausibility**: Distractors are assessed on a 3-point scale. 0 points: The distractor is obviously wrong. 1 point: The distractor is somewhat confusing. 2 points: The distractor is highly confusing, making it difficult to determine the correct answer.

The judges were presented with the cloze questions, where each cloze question consists of a stem, the correct answer, and three distractors, together with the definition of the correct answer. Table 2 shows the evaluation results.

Table 2: Human valuations of the quality of the generated distractors

	Reliability	**Plausibility**
Judge 1	0.9600	1.7300
Judge 2	0.9878	1.6922
Judge 3	0.9801	1.7234
Avg	0.9760	1.7152
Stdev	0.0117	0.0164

The high average reliability score of 0.976 indicates that the vast majority of distractors are contextually appropriate and grammatically correct, yet logically incorrect in the context of the question to be true distractors. The average plausibility score of 1.7152 is closer to 2 than 1, which indicates that the distractors are sufficiently confusing, making it challenging for examinees to immediately identify the correct answer.

These results further demonstrate that AI-WordNet meets the requirements for generating high-quality distractors in the creation of cloze questions.

5.3 Predicted and True Hypernyms

If the predicted hypernym does not align with the true hypernym, the predicted hypernym may still suffice for generating distractors in cloze questions. In such cases, the primary requirement is for the distractors to create enough confusion within the same contextual space, rather than achieving perfect linguistic accuracy. Below are several examples of lemmas, synset names and the corresponding glosses extracted from PWN 3.0, accompanied by explanations that demonstrate why the predicted hypernym, despite not being perfectly accurate, still leads to effective distractor generation.

(1) For the synset ‘diabetes.n.01’ with the gloss ‘a polygenic disease characterized by abnormally high glucose levels in the blood; any of several

metabolic disorders marked by excessive urination and persistent thirst', the true hypernym is 'polygenic_disorder.n.01'. AI-WordNet generates 'metabolic_disorder.n.01' as its hypernym synset, which aligns with the characteristic of diabetes as a metabolic disease. Thus, it is deemed adequate for generating distractors.

(2) For the synset 'seven_seas.n.01' and the gloss 'an informal expression for all of the oceans of the world', the true hypernym is 'body_of_water.n.01'. AI-WordNet generates 'ocean.n.01' as its hypernym synset, which is deemed adequate for generating distractors, although it should logically be considered a hyponym rather than a hypernym.

(3) For the synset 'cold_turkey.n.01' and the gloss 'a blunt expression of views'[2], the true hypernym is 'expression.n.03'. AI-WordNet generates 'opinion.n.01' as its hypernym synset, correlating to the expression of views to some extent and so is deemed adequate for generating distractors, although it doesn't capture the full essence of the term.

(4) For the synset 'south_pacific.n.01' and the gloss 'that part of the Pacific Ocean to the south of the equator', the true hypernym is 'part.n.03'. AI-WordNet generates 'pacific.n.01' as its hypernym synset, correctly recognizing the part-whole geographic relationship, and so is deemed adequate for generating distractors.

(5) For the synset 'company.n.09' and the gloss 'a unit of firefighters including their equipment', the true hypernym is 'unit.n.03'. AI-WordNet generates 'fire_department.n.01' as its hypernym synset, which is contextually relevant in the firefighting context. Thus, it is deemed adequate for generating distractors, even though it fails to convey the concept of 'company.n.09' as a unit.

These examples demonstrate why using AI-WordNet in generating distractors achieves high reliability and plausibility, even though AI-WordNet only achieves 64.8% exact match of the hypernyms in PWN 3.0.

On the other hand, the current implementation of AI-WordNet may generate hypernyms for certain short glosses that are not contextually appropriate. For example, for the synset 'pipa.n.01' with gloss 'type genus of the Pipidae', AI-WordNet outputs 'bird_genus.n.01' as its hypernym, which is entirely incorrect. This issue likely arises due to the brevity of the gloss, where the only available information is the term 'Pipidae' without further description. As a result, the underlying models fail to fully grasp the meaning. Such issues can be mitigated by enhancing the knowledge base of the underlying models.

5.4 Hypernym Extraction vs. Generation

As mentioned in Section 4.5, over 80% of glosses in WordNet-N contain inherited hypernyms. Exploring how to directly identify and extract these hypernyms from the glosses faces the following challenges: (1) Distinguishing between glosses that contain inherited hypernyms and those that do not is challenging. Glosses may not explicitly mention their hypernyms, making it hard to categorize them accurately. (2) Even if we can determine that a gloss contains an inherited hypernym, extracting it is not straightforward. Glosses are often abstract or formulated in a way that doesn't explicitly convey the hypernym. For instance, the gloss for the synset 'idleness.n.01' is 'having no employment', but its hypernym synset is 'inactivity.n.03'. Extractive methods might mistakenly identify 'employment' or 'no employment' as the hypernym, which is incorrect.

Overcoming these challenges requires more sophisticated approaches, potentially combining extractive techniques with deeper semantic analysis to capture the correct hypernyms.

We note that certain hypernyms predicted by AI-WordNet, which do not align with the ground-truth hypernyms, could potentially be resolved more effectively through extractive methods. For example, for the synset 'adenopathy.n.01', its gloss is 'a glandular disease or enlargement of glandular tissue', and its true hypernym is 'glandular disease', which is explicitly mentioned in the gloss. AI-WordNet, however, predicts 'pathology.n.01' as its hypernym synset, which, although contextually acceptable for generating distractors, is not as accurate as an extractive approach.

This observation suggests that integrating both generative and extractive methods may offer substantial improvements in hypernym prediction, warranting further investigation.

6 Conclusions and Final Remarks

AI-WordNet represents our initial effort to expand existing lexical datasets based solely on glosses,

[2] This lemma has another synset 'cold_turkey.n.02' in PWN 3.0 with gloss 'complete and abrupt withdrawal of all addictive drugs or anything else on which you have become dependent'.

aiming to address the specific requirements of generating controllable and explainable distractors for cloze questions. This approach is particularly useful in today's rapidly evolving lexicon, where new lexemes frequently emerge. We demonstrated the utility of AI-WordNet in generating high-quality distractors for cloze question generation from a given text.

Similarly, AI-WordNet can be applied to create controllable and explainable distractors for various types of multiple-choice questions within the framework of AI-oracle machines. AI-oracle machines use a combination of PLMs as oracles and conventional algorithms to perform complex tasks by breaking them into manageable subtasks, guiding query formulation, and ensuring alignment with predefined requirements (Wang, 2024). By integrating AI-WordNet into this framework, it becomes possible to generate distractors that are not only contextually relevant but also transparent in their derivation, thereby enhancing the quality and reliability of multiple-choice assessments.

Moreover, AI-WordNet can be used to create a specialized WordNet from scratch for lemmas in a specific domain, such as a particular area of medicine or a new field in the sciences, as long as they are in the same language as the models trained on the existing WordNet and their glosses are available. Whether expanding an existing WordNet or creating a new one from scratch, AI-WordNet can be used to provide an initial version for domain experts to refine and build upon.

For an under-resourced language lacking an extensive WordNet, such as PWN 3.0 for English, AI-WordNet can be employed to construct a version of WordNet within the framework of transfer learning, based on the premise that all human languages share a common root set of synsets, as represented in PWN 3.0. This can be achieved by identifying a moderate number of critical synsets unique to the language, along with their lexical names and hypernym-hyponym relations, and using this information to fine-tune models initially trained on PWN 3.0. Note that this gloss-based approach can be used to create a parallel language with a different vocabulary, which might be useful for generating a secret code to transmit messages, similar to the Code Talker project undertaken by the US military during WWII, which used the Navajo language spoken by Native American tribes for secure communications.

To improve the direct-hit ratio of the predicted hypernyms, we may explore the combination of generative and extractive methods, enhance the underlying PLMs and devise new gloss-sense discerning techniques that go beyond traditional word-sense disambiguation methods.

As a closing remark, we experimented with replacing the underlying PLMs with GPT-3.5-Turbo, out-of-the-box and fine-tuned. Unfortunately, we find that this approach performs significantly worse on our tasks compared to specialized smaller models. This finding demonstrates that for certain tasks, properly fine-tuned smaller models can deliver better performance while being more cost-effective.

Acknowledgment. This work was supported in part by Librum Technologies, Inc.

References

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *Preprint*, arXiv:2111.01998.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975.

Yicheng Sun and Jie Wang. 2023. Generate cloze questions generatively. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2023)*. IEEE and the International Neural Network Society.

Koeva Svetla. 2021. Towards expanding WordNet with conceptual frames. In *Proceedings of the 11th Global Wordnet Conference*, pages 182–191, University of South Africa (UNISA). Global Wordnet Association.

Jie Wang. 2024. AI-oracle machines for intelligent computing. *Preprint*, arXiv:2406.12213.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *Preprint*, arXiv:2310.07554.