Lexicography in NLP: A Study on the Interaction Between Lexical Resources and Large Language Models

A thesis submitted in partial fulfilment

of the requirement for the degree of Doctor of Philosophy

Fatemah Almeman

August 2025

Cardiff University
School of Computer Science & Informatics

Abstract

This thesis explores the interaction between lexical resources (LRs) and large language models (LLMs) in the context of natural language processing, focusing on the evaluation of WordNet (WN)—the de facto lexical database for English—along with the development of a new dataset and a novel reverse dictionary (RD) method. The investigation starts with an assessment of WN, particularly its examples, both intrinsically and extrinsically, compared to other resources using the Good Dictionary EXamples (GDEX) framework. This evaluation shows that WN's examples are often limited in length and informativeness. In an extrinsic analysis, we examined WN's performance in definition modeling and word similarity tasks, where informative contextual representations are essential. Results indicate that LLM-generated examples are more informative than those from WN.

To overcome limitations in LRs (some uncovered by our analysis), we then introduce a new dataset called `3D-EX` providing terms, definitions, and usage examples. It integrates entries from ten diverse English dictionaries and encyclopedias with varying linguistic styles. We conducted intrinsic experiments on source classification, predicting the origin of a `<term, definition>` instance, and RD, which retrieves a ranked list of terms from a definition. Results indicate that 3D-EX enhances performance in both tasks, highlighting its usefulness for NLP.

This thesis further explores RD by introducing GEAR, a lightweight and unsupervised approach to RD tasks. GEAR operates through four stages: Generate, Embed, Average,

and Rank. It was evaluated using the Hill dataset, a leading benchmark for RD tasks, and it consistently outperformed existing methods.

In conclusion, this thesis investigates how LLMs and LRs can benefit each other. We identified limitations in some resources and found that LLMs are a suitable tool for addressing them. Additionally, LLMs can automatically improve language resources by unifying them with different anchors. Datasets and code are publicly available.

Contents

List of Figures

List of Tables

List of Publications

The work introduced in this thesis is based on the following publication:

1. Fatemah Almeman and Luis Espinosa Anke. 2022. **Putting WordNet's Dictionary Examples in the Context of Definition Modelling: An Empirical Analysis**. In Proceedings of the Workshop on Cognitive Aspects of the Lexicon, pages 42–48, Taipei, Taiwan. Association for Computational Linguistics. [6]

2. Fatemah Almeman, Hadi Sheikhi, and Luis Espinosa Anke. 2023. **3D-EX: A Unified Dataset of Definitions and Dictionary Examples**. In Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, pages 69–79, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria. [8]

3. Fatemah Almeman, Steven Schockaert, and Luis Espinosa Anke. 2024. **WordNet under Scrutiny: Dictionary Examples in the Era of Large Language Models**. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 17683-17695, Torino, Italia. ELRA and ICCL. [9]

4. Fatemah Almeman and Luis Espinosa Anke. 2025. **GEAR: A Simple GENERATE, EMBED, AVERAGE AND RANK Approach for Unsupervised Reverse Dictionary**. In Proceedings of the 31st International Conference on Computational Linguistics, pages 8242–8254, Abu Dhabi, UAE. Association for Computational Linguistics.[7]

Acknowledgments

First and foremost, I am deeply thankful and blessed to have my supervisor, Luis Espinosa-Anke, for all your support even before I started my PhD. Your warm, welcoming kindness and belief in me were very powerful for me as a PhD student and prepared me mentally to proceed further in my studies. Your endless support encouraged me to work hard and progress in my studies with joy and confidence, despite the challenges. You taught me how to be a good researcher, academic, and scientist through your daily guidance — which greatly developed my skills. I am grateful for your confidence in me and for always pushing me to go as far as I could in my research.

Also, I would like to extend my deep thanks to my second supervisor, Professor Steven Schockaert. Your suggestions have always improved my research. I also want to thank the Cardiff NLP Group for being so awesome and helpful, and for funding the user study (£1000), which greatly supported this research.

Importantly, I need to thank the source of my funding, as this PhD work would not have been possible without their financial support. I am extremely grateful to the Kingdom of Saudi Arabia Government for providing me with a full scholarship throughout my PhD program under the academic faculty staff of Princess Nourah bint Abdulrahman University.

Finally, this work would have never been completed without the continuous support and patience of my parents. Thanks to my father and mother for believing in me and for your endless support from my childhood to my PhD. Thanks to my sisters and brothers

for your encouragement and support throughout my PhD journey. Many thanks to you, my husband – you stood by me throughout all the challenges I faced and helped me overcome them. Also, thanks to my sweetheart, my daughter, for always making me smile despite my difficulties.

Chapter 1

Introduction

1.1 Background and Motivation

Natural language Processing (NLP) is the field that explores how computers can interact with humans by understanding and manipulating natural language [45]. Different technologies and applications we use in our daily life are based on NLP. For instance, machine translation applications, which generate the best possible translation without human assistance, like Google translate [190]; chatbot applications, which simulate human conversation for purposes such as customer service and personal assistance [50]; speech recognition software, which recognizes patterns in speech like Apple's Siri and Amazon's Alexa [170]; or e-mail phishing detection, which filters mailboxes from unwanted junk mails [141]. Most of the current NLP technologies are based on Language Models (LMs). These models use machine learning to predict the probabilities of future or missing words by examining word distributions [226].

Initially, LMs are *pre-trained* on a large amount of text data. During this pre-training phase, the model learns word representations, which are defined as "a mathematical object associated with each word, often a vector. Each dimension's value corresponds to a feature and might even have a semantic or grammatical interpretation" [200]. These learned representations are then used in supervised training for downstream tasks, with optional fine-tuning of both the representations and the network from the initial (pre-training) stage [84]. Fine-tuning involves adjusting the parameters of a pre-trained

language model to fit a specific task or domain without the need to train the model from scratch. For example, a LM can be fine-tuned using labeled datasets for text classification, where texts are automatically categorized based on predefined labels [36]. Fine-tuning pre-trained LMs such as BERT [53], RoBERTa [126] or GPT-2 [167] is common to get high-quality lexical and sentence representations [123, 124], as they have been shown to generate sub-optimal embeddings when used directly, without additional task-specific fine-tuning [117, 68]. In this context, LMs learn to perform well on a given task or dataset by observing past examples, which are typically manually labeled or assigned *soft* labels derived from metadata, geographical information, etc. [161, 146]. A further development in LMs is the emergence of large language models (LLMs), which are significantly larger in size and number of parameters. LLMS such as LLaMA [198] and GPT-4 [2] are capable of handling more complex and diverse language tasks with greater efficiency [86]. LLMs, commonly associated with text generation tasks, can produce high quality text that it is sometimes preferred over human-generated content [213, 33]. Unlike the LMs discussed earlier, LLMs are much larger and have stronger language skills [140], but this comes with drawbacks such as slow training and inference, high hardware requirements, and increased operational costs [147]. As a result of these limitations, efforts have been made in creating more efficient architectures [44, 223], and better training strategies [172, 169].

Due to the fact that LMs and LLMs learn from diverse, often noisy sources like web content, this can cause biases and incorrect information. Using Lexical Resources (LRs) provides structured data and high-quality lexical information, which is important for improving models' understanding of language and their performance in NLP tasks. LRs serve as a fundamental repository of knowledge, containing different details about words, including definitions, usage examples, morphology (the forms of words), syntax, and etymology (the origin and history of a particular word). These features make LRs a fundamental tool for learning new languages because they help learners to enhance their vocabulary, word usage, and more generally, improve their language skills. The relationship between LRs and NLP is strongly interdependent, with each

enhancing the capabilities of the other. Providing valuable information about word meaning, syntactic structures, and semantic relationships empowers NLP systems to analyze, process, and generate human-like texts accurately and efficiently [34]. What follows are some examples of the interplay between NLP and LR.

LRs for NLP For example, in the Word Sense Disambiguation (WSD) NLP task, which aims to map an ambiguous (multi-sense) word in a context to the correct sense [171] and where the de-facto sense inventory in WSD is WordNet [135], the use of LRs to obtain more context can assist in disambiguation. For instance, consider these two examples:

- (a) *I can hear bass sounds,*
- (b) *They like grilled bass.*

The word *bass* clearly shows different meanings—referring to low-frequency tones in example (a) and a type of fish in example (b) [148]. Therefore, context or usage examples provided in LRs can effectively help resolve ambiguity. It is also important to highlight that usage examples are more effective than random mentions of terms in sentences because they are intended to be informative, easy to understand, and helpful for learning definitions [108].

Additional instances of using LRs in NLP are in information retrieval systems, which involve finding documents that match a user's query from among a large collection of documents [204]. Here, LRs assist in understanding the meaning and context of search queries, enabling more accurate results. Moreover, in information extraction tasks, such as named entity recognition, which is the process of identifying the names of organizations, people, locations, or other entities in text [130], and entity linking, which involves connecting mentions of entities in text to their metadata, such as synonyms or translations in a knowledge base, which often consists of a collection of LRs [188].

NLP for LRs Conversely, NLP techniques have also been effectively used in creating and improving LRs. For instance, updating LRs for newly emerged terms requires a collaborative effort by a team of domain experts. NLP tasks such as definition extraction, which involves finding meanings in sentences automatically, and the automatic extraction of examples can be easily done to include definitions and usage examples for these new terms [151, 24, 108]. Enriching LRs with lexical collocations that should greatly aid second language learning by automatically extracting these combinations from corpora [59, 61]. Additionally, Velardi et al. [203] and Alfarone and Davis [5] applied techniques to automatically extract hypernymy relations[1] from text corpora.

Despite the range of applications and interactions between NLP and LRs mentioned above, their effectiveness is strongly dependent on the quality of the LRs under consideration. To properly evaluate and understand the quality of these resources, it is necessary to conduct in-depth evaluations. However, performing these evaluations, while essential, is also not trivial due to the absence of gold standard for comparison, as different resources may have different scopes. Additionally, defining a single evaluation metric that captures all aspects of the resource's quality and usefulness across different tasks is difficult [18]. Evaluating LRs could be intrinsic, focusing on the quality of the lexical resource itself, independent of specific NLP tasks such as evaluating their lexical coverage [165] and the connectedness between terms in taxonomy-based LRs (which organize terms in hierarchical structures) [26], or extrinsic, measuring how well an NLP system performs with the resource in a given task [195].

Motivation: This thesis will investigate the interplay between particular elements in lexicography (dictionary terms, definitions and examples) and NLP. We first aim to evaluate LRs, such as WordNet—the primary LR used in NLP applications—from different perspectives, and identify their limitations or issues. Current resources often lack completeness, as they fail to provide example usages for some terms, and inform-

[1] A semantic relation where one word is a general category (hypernym) that includes more specific words (hyponyms). For example, "vehicle" is a hypernym of "car.".

ativeness, as the provided examples may lack usefulness. Additionally, by exploring advanced models like ChatGPT, we aim to generate more informative and reliable resources. Moreover, developing a homogeneous and comprehensive dataset from different resources can solve the issues of inconsistency and provide a rich base for NLP tasks.

1.2 Hypothesis and Research Questions

Dictionary examples serve as essential tools for elucidating the contextual and semantic meanings of terms. Within the scope of lexical resources, these examples play an important role in providing clarity and depth to language understanding. The main hypothesis underpinning this thesis is that the quality of dictionary examples significantly influences the performance of NLP models that rely on word- and sentence-level semantics. Specifically, it suggests that when dictionary examples are detailed and informative, they empower NLP models to better understand term usage and meanings.

In order to verify this hypothesis, the following research questions are addressed:

Research Question 1: How does the quality and length of dictionary examples within LRs, such as WordNet, impact the performance of NLP models in tasks such as definition modeling?

Research Question 2: How can embeddings for words, phrases, and sentences be improved by leveraging dictionary examples?

Research Question 3: How does the integration of multiple lexical resources into a centralized knowledge repository contribute to improving NLP models across different downstream tasks?

Research Question 4: Can the task of reverse dictionary improve by combining "the best of both worlds", i.e., LLM generation capabilities as well as semantic similarities derived from dictionary embeddings?

1.3 Contributions

- We perform an in-depth evaluation on WordNet, and find that despite its proven quality and adoption, it also does not seem to be an optimal resource when informative contexts (i.e., dictionary examples) are required for downstream tasks such as definition modeling or deriving contextualised representations. These findings were later supported by Giulianelli et al. [81].
- We introduced 3D-EX, a dataset that combines different English resources into one, providing <term, definition, example, source> sets. This dataset aims to serve as a standardized benchmark for lexical semantics tasks. *3D-EX* has already proven valuable as a source for LM pre-training, as demonstrated by Gajbhiye et al. [76].
- We proposed GEAR (**g**enerate, **e**mbed, **a**verage and **r**ank), a state of the art method for Reverse Dictionary (RD) that utilizes an LLM for generating a set of candidates given an input definition, and pools their corresponding embeddings into a vector used for KNN search.

1.4 Thesis Structure

- Chapter 2 – Background and Related Work – provides an in-depth review of the interplay between LRs and NLP, and some of the other related areas.
- Chapter 3 – WordNet under Scrutiny: An Empirical Analysis – presents intrinsic and extrinsic evaluation of of the well-known lexical database WordNet, focus-

ing specifically on its dictionary examples. We assess these examples both directly, by comparing them against criteria for well-constructed dictionaries, and indirectly, through NLP tasks.

- Chapter 4 – 3D- EX : A Unified Dataset of Definitions and Dictionary Examples – introduces 3D-EX, a centralized repository that unifies a diverse set of English dictionaries and encyclopedias. It can be used to train and test definition modeling systems, explore out-of-domain generalization, and, most importantly, act as a unified test bed for lexical semantics tasks.

- Chapter 5 - GEAR: A Simple GENERATE, EMBED, AVERAGE AND RANK Approach for Unsupervised Reverse Dictionary - proposes a simple approach to RD tasks that leverages LLMs in combination with embedding models. Despite its simplicity, this approach outperforms supervised baselines in well studied RD datasets, while also showing less over-fitting.

- Chapter 6 - Conclusion and Future Work - concludes the thesis by summarizing our contributions and findings, while also highlighting potential areas for future research.

1.5 Summary

In this chapter, we have introduced the background and the motivation to work on the considered topic. We also discussed the hypothesis, the main research questions, the thesis contributions and structure. The next chapter will provide more detailed background information and focus on the literature of the considered topic.

Chapter 2

Background and Related Work

2.1 Introduction

This thesis aims to explore the interplay between LRs and NLP. LRs play a crucial role in various NLP tasks, enhancing the performance and accuracy of systems across different applications. This chapter will present the required background knowledge about LRs and their main components, in addition to reviewing existing research on the integration of LRs into NLP systems.

The chapter is divided into two main sections. Section 2.2 provides a general overview of LRs, explaining what they are, showing a few examples, and then detailing WordNet as the primary source used in NLP in Section 2.2.1. The main components in LRs are discussed then in Section 2.2.2 and 2.2.3. Section 2.3 examines how LRs are used in different areas of NLP, with a focus on how each component (definitions or examples) contributes to various NLP tasks. Finally, this thesis examines two NLP tasks: Word Similarity and Reverse Dictionary, which are covered in Section 2.3.3.

2.2 Lexical Resources

A LR is a database that may consist of one or more dictionaries, depending on its language scope [181]. LRs describe words and relations between them by providing

essential lexicographic and contextual information such as definitions, examples, and translations. Dictionaries are the primary example of LRs and have long served as essential tools for both language learners and teachers, as they provide the fundamental information needed to understand word meanings [207]. They can be monolingual or bilingual, and may be intended for general audiences or specialized domains.

Some common examples of lexical resources include WordNet (WN) [139], which is discussed in detail below, as it is the primary resource used in this thesis. WN is an electronic lexical dictionary for English that organizes words into groups of synonyms [139]. While WN is the focus of this thesis and, therefore, will receive most attention in this chapter, it is important to put it in the context of other analogous resources. For instance, ConceptNet [98] is a key resource for commonsense reasoning, represented as a directed graph where nodes are concepts and edges are assertions that express relationships between them [24, 180]. BabelNet [149], on the other hand, is a large multilingual semantic network that links Wikipedia, the largest multilingual web encyclopedia, with WN [150].

2.2.1 WordNet

WN is one of the most well-known lexical resources in NLP and has been described by Hovy et al. [94] as having a "seemingly endless list of papers using it." Unlike traditional dictionaries, which typically cover a broad range of word classes, WN focuses on content words — nouns, verbs, adjectives, and adverbs — comprising over 155,000 words organized into approximately 117,000 synsets (synonym sets). These include around 82,000 noun synsets, 13,800 verb synsets, 18,200 adjective synsets, and 3,600 adverb synsets [1]

[1] All statistics are based on WordNet version 3.1: `https://wordnet.princeton.edu/`. Synsets in WN organize words into groups of synonyms and are described by their definition, lemmas[2], examples, and the relations they have with other synsets, such as hypernymy (is-a), meronymy (is-part), troponymy (manner-of), etc. While WN has seen some use in lexicography and language learning [142],

For example, WN is the primary sense inventory used in WSD tasks, as discussed in Chapter 1, due to its synset structure and semantic relationships, which provide the context and hierarchy needed to resolve word ambiguity. Disambiguating polysemous words involves collecting related terms from synsets, glosses, and various hypernym levels, which are then compared using different WSD algorithms [54]. Additionally, KnowBert [163] integrates WN into BERT to enhance its word embeddings by linking words to their corresponding senses. The results show that BERT enriched with WordNet outperforms the original BERT-large model.

Recent studies show that WN remains essential in various NLP applications. In 2025 alone (as of May 8, 2025), more than 1,800 published papers have mentioned WN, according to Google Scholar. For instance, Melacci et al. (2024) [133] improved supervised WSD models by integrating semantic features from the WN and WordNet domains, resulting in better performance in standard benchmarks. Similarly, Wenjun et al. (2024) [212] proposed a WSD method based on multiple sense graphs, combining BERT embeddings and PageRank algorithms to determine the most appropriate word senses. Additionally, WN has been applied in semantic similarity tasks, such as identifying relevant examination questions, as demonstrated by Goh et al. (2023) [82]. In multilingual NLP, efforts to enrich Arabic WN using machine translation and transformer models, improving its use in different linguistic contexts [73, 191].

However, WN has several limitations, including its focus on only English, its static nature, and the absence of important linguistic relations like word combinations. Additionally, its synsets often have very specific and narrow categories, which can make it difficult to use in some tasks [67]. To overcome these issues, different methods have been proposed, such as including more synsets and senses from other resources since the English language is not static and there is always a need to update WN [131]. Collocational information has also been added to WN relations in previous work, which is essential in some tasks like Machine Translation [57], where word-by-word transla-

its primary application has been in computational linguistics and NLP, as will be shown throughout this thesis.

tions often fail to capture the meaning of collocations and can lead to errors [186, 122]. Additionally, several research works have been conducted to generate translations for WN in several languages [25].

WN and other LRs differ in their components based on specific needs. However, some key elements remain essential across most LRs, including definitions and examples. Definitions provide the meaning of lexical items, while examples illustrate their usage in context. The following sections will provide a detailed explanation of these components.

2.2.2 Dictionary Definitions

Definitions are the main content of dictionaries to help understand the meaning of unknown terms [158]. Definitions should be accurate and clear, and complete enough to meet the reader's needs and to differentiate between different related meanings or senses [144]. However, several problems associated with dictionary definitions in general need to be considered. For example, circularity where the defined term is shown in the definitions [32] such as this definition "*Happy: feeling happiness.*" Obscurity is another serious issue where definitions are complex or use obscure language, making them difficult for users to understand such as this definition of sand where some parts of it (like comminuted and silicious) are harder to understand "*Sand: a material made of crushed rock fragments and smooth particles, mostly containing silica, finer than those in gravel .* [3] *" [3].* We will observe these issues practically in Chapter 3.

Definitions typically consist of at least three components which are the definiendum (the term being defined), the definiens (what defines the term), and a connector (like a verb or punctuation) that links them together [206]. Based on the connectors, Westerhout 214 found that there are four common definition types: the first uses 'to be' as the connector; the second uses other verbs or verbal phrases (e.g., to mean, to comprise);

[3]Oxford English dictionary online. [www.oed.com]

the third uses punctuation marks like colons; the fourth uses pronouns to refer back to a term mentioned earlier, followed by the definition. Within a definiens, usually there is a term that is closely related to the definiendum and is often a synonym or hypernym and it is called *genus term*. For example, in *"Automobile: a motorcar"* motorcar acts as a synonym. Similary, in *"Summer: the second and warmest season of the year"* season is the hypernym (genus term) of summer [208]. Identifying genus terms helps to process definitions, which often follow consistent patterns in syntax, style, and vocabulary [10].

The above considerations about definition typification and quality assessment are relevant to this thesis because, as we will show, definitions in widely used resources such as WN are not always optimal. This limits their usefulness for various types of end users. While WN was initially developed for natural language processing (NLP) systems, it has also been applied in language learning, where it helps learners distinguish between near-synonyms and similar-looking words that are often confused [194], and supports the teaching of lexical semantics and semantic relations between English nouns [164]. Moreover, its integration into structured knowledge bases such as YAGO [173] and DBpedia [13], both of which incorporate lexical information from WN, has further expanded its reach. These resources are used by terminologists and digital lexicographers to organize lexical content and support tasks such as terminology management and sense alignment, where limitations in definitions can reduce the accuracy of terminology and make it harder to match terms correctly across languages [196, 132]. Similarly, knowledge engineers rely on these resources to develop ontologies and reasoning systems, which require clear and consistent definitions to make correct inferences and classifications [155, 83].

2.2.3 Dictionary Examples

Using examples systematically to explain the English words that are defined in the dictionaries was used first by Samuel Johnson in his dictionary *A Dictionary of the*

English Language [101], and the examples were taken from professional authors' writings like Shakespeare and Addison [49]. Besides showing words in context, examples clarify meanings of words with difficult definitions—multiple corpus examples can sometimes be more useful than the definition itself [12]. They also aid in navigating long lists to find specific senses by matching the example to the correct sense [90]. As stated by Frankenberg-Garcia [70], "an entry can be almost incomprehensible without its examples".This underscores the importance of examples in dictionary definitions. According to Atkins and Rundell [12], a good dictionary example must be:

- **Typical**, i.e., showing the, as Kilgarriff et al. [108] put it, "frequent and well-dispersed patterns of usage" of the target word that represent various contexts and usages of the word across different situations.
- **Natural**: the example should appear like a sentence one would expect to see in usual language use.
- **Informative**: so that it helps with understanding the definition of the word.
- **Authentic**: because they are examples from actual corpora.
- **Self-contained**: the content of the example is understandable without requiring additional context.
- **Intelligible** to the reader by avoiding difficult lexis and structures which cannot be understood without access to a wider context (a.k.a. *readability*).

2.2.3.1 GDEX

One of the most prominent contributions in dictionary examples is GDEX, which stands for Good Dictionary EXamples. It is a system that added around 8,000 new example sentences to the Macmillan English Dictionary by automatically finding good examples in corpora using a set of rules of thumb [108]. GDEX is motivated by the fact

that creating example sentences manually is time-consuming and increases the cost of developing lexicographic resources [89]. Other researchers have also explored automating the selection of example sentences, such as Frankenberg-Garcia [71], who used large English corpora (e.g., BNC, COCA, and UKWaC) to provide helpful context for understanding and using specific words.

GDEX has translated the good dictionary features into practical metrics such as sentence length, word frequencies. For instance, sentence length affects informativeness and readability: very short sentences may lack context, while very long sentences can be harder to read and understand. These metrics are later used as a method to evaluate and measure the quality of examples against GDEX criteria [134].

GDEX Applications GDEX was initially developed as a feature in the Sketch Engine[4] [107], a tool designed to assist lexicographers in identifying high-quality dictionary examples by analyzing various lexical and syntactic characteristics, such as example length and complexity. It has since been used in various projects, including ColloCaid [72], a writing assistant tool focused on providing academic English collocation suggestions. GDEX aids in the example selection process in ColloCaid by applying automated penalties to concordances with long sentences, rare words, excessive capital letters, non-alphanumeric characters, or anaphoric references. Another notable work utilizing GDEX is the DANTE (Database of Analysed Texts of English) project [106, 100] [5]. In DANTE, GDEX was employed to streamline the selection of example sentences for dictionary entries. By applying various filters such as sentence length, the presence of rare words or proper names, and the number of pronouns, GDEX effectively ranked sentences, allowing lexicographers to identify suitable examples more efficiently. Figure 2.1 shows a screenshot of DANTE interface.

[4] http://the.sketchengine.co.uk

[5] `https://github.com/lexicalcomputing/dante`

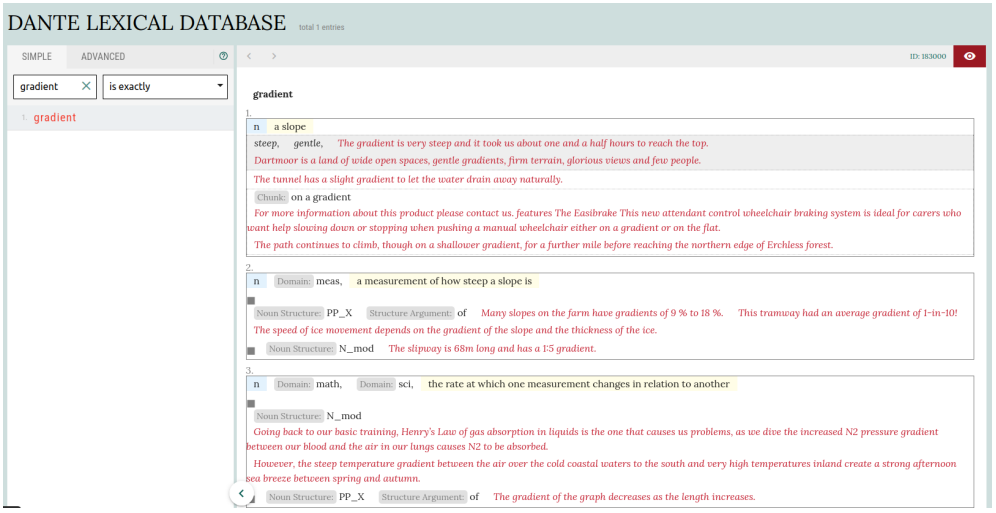


Figure 2.1: Screenshot of the Dante interface showing examples of the word gradient.

2.3 Lexical Resources and Modern NLP

Lexical resources (LRs) in general, and dictionaries in particular, have played a critical role in recent years in enhancing both knowledge-rich and organically derived NLP systems. For example, DictBERT integrates structured entries from the Cambridge Dictionary, including definitions, synonyms, and antonyms, to enrich language models with lexical knowledge [43]. Similarly, Unified Reversible Definition Modeling leverages tuples from the Oxford English Dictionary containing a word, its usage, and its definition to build a neural dictionary that supports both retrieving words from definitions and generating definitions for given words [42]. Faruqui et al. [65] retrofitted word embeddings, using semantic relations in PPDB [78], WN, and FrameNet [17], by presenting a method to improve vector space representations by using relational information from semantic lexicons. It encourages words that are linked in the lexicons to have similar vector representations. Joshi et al. [102] used definitional information to augment pre-trained Language Models (LMs) by proposing a technique for representing input texts by embedding them in context with dynamic textual knowledge retrieved from various documents. This method is applied to reading comprehension

tasks, where questions and passages are encoded with background sentences related to the mentioned entities. Delli Bovi et al. [52] and Xu et al. [217] used definitions for generating knowledge bases. Delli Bovi et al. [52] introduce DEFIE, a method for extracting information from a large number of text definitions by analyzing the structure and meaning of these definitions. Espinosa-Anke et al. [58] propose EXTASEM! which is a method for automatically creating lexical taxonomies by collecting definitions, extracting (hyponym, hypernym) pairs, and finally building a taxonomy with weighted connections based on domain relevance. Xu et al. [217] introduce Taxo-Prompt, a framework that uses prompts to learn global structure with taxonomic context. It improves semantic understanding by using a modified random walk algorithm to enhance prompts. In the following two sections, we discuss how dictionary components, specifically definitions and examples, are used in NLP tasks.

2.3.1 Dictionary Definitions and Language Models

Definitions are a fundamental building block in lexicography, linguistics and computational semantics. In NLP, they have been used for retrofitting word embeddings or augmenting contextual representations in language models. However, LRs containing definitions vary in structure, style, and coverage, which can impact the performance of models trained on them. For example, WN is widely used in definition modeling, but its definitions are designed to be short and consistent, which can limit the quality of the model [21]. To better understand how definitions are applied in NLP and why their quality matters, we now show several tasks where definitions play a key role.

Definition Modeling (DM): DM, as introduced by Noraset et al. [156], is the task of generating a dictionary definition for a given word. This task was made possible by the adoption in NLP of sequence-to-sequence architectures based on RNNs [79]. Recently, DM systems have shown impressive performance in several intrinsic and downstream tasks, mostly thanks to being able to go from context-less (Noraset et al.

only used the definiendum[6] as a conditioning token at all timesteps) to a contextually richer setting, e.g., by conditioning the generated definition to an example of usage of the target word. Ni and Wang [154] present a method for automatically explaining new, non-standard English expressions using data using a neural model with dual encoders: one for context and one for the expression itself. This allows us to generate accurate definitions based on context. Gadetsky et al. [74] developed a technique to evaluate word vectors by modeling dictionary definitions and addressing word ambiguities using latent variable modeling and soft attention mechanisms. Given a context and a target word, Chang et al. [38] algorithm transforms the word into a sparse vector, selects dimensions to capture its meaning, and uses an RNN to generate a human-readable definition. Zhu et al. [227] studied Multi-sense Definition Modeling, using sequence-to-sequence neural networks to generate definitions for each sense of pre-trained word embeddings. Ishiwatari et al. [99] used a variant of the encoder-decoder model that captures the local context (explicit contextual information in a sentence) with the encoder and the global context (implicit information from large text corpora) with the decoder, initialized by the target phrase's embedding.

A notable leap in DM was achieved in Bevilacqua et al. [21], who fine-tuned BART [116] on example-definition pairs, and reported high results in intrinsic benchmarks and, more importantly, used their DM system for downstream NLP, specifically WSD and word-in-context classification. DM has also been explored from other perspectives, e.g., generating definitions with appropriate specificity using re-ranking mechanisms to solve over/under-specificity problems[96], defining scientific terminologies and controlling the complexity of generated definitions based on reader's background knowledge [14], combining extraction and generation, incorporating a web-based extraction method into the generation process for jargon [7] definition modeling [97], or extending the generation cross-entropy loss with a reconstruction objective [110] (re-

[6]The *genus-et-differentia* Aristotelian definitions follows an *A is a B which Z* structure, with *A* being the *definiendum*, *B* the *genus* and *Z* the *definiens* or *differentia specifica*.

[7]A specialized term used by experts within a specific field.

miniscent of works that used dictionary definitions for improving word embeddings via autoencoders [27] or LSTMs [93]).

Definitions in LMs pre-training: While Pre-trained Language Models (PLMs) have achieved the state-of-the-art performance across different NLP tasks, they struggle with knowledge-driven tasks. DictBERT [43] introduces a novel approach to enriching PLMs with dictionary knowledge. This method involves two pre-training tasks: predicting dictionary entries, using a description to predict its masked entry and learn entry representations from descriptive texts; and entry description discrimination, which improves the robustness of entry representations through contrastive learning, which constructs positive and negative samples using dictionary synonyms and antonyms. In Dict-BERT [222], PLMs are enhanced by incorporating definitions of rare words from dictionaries during pre-training. This is important since word representation quality depends on its frequency and rare words often have poor embeddings. In natural language understanding (NLU), CoDA21 (Context Definition Alignment) is a challenging benchmark that evaluate the natural language understanding abilities of PLMs [184]. The task involves providing a definition and a context for k words, without knowing the words themselves, and the challenge is to align the k definitions with the k contexts. The results show a large gap between human and PLM performance, highlighting CoDA21's importance as a benchmark for improving NLU capabilities in model design.

Definitions for improving word embeddings: Word embedding models are approaches for learning dense vector representations of words within a continuous vector space. They effectively capture semantic and syntactic relations and the vector representations learned through these models have shown high performance across various tasks such as information retrieval [77, 152] and sentiment analysis [221]. Traditional word embedding techniques such as word2vec [138] and GloVe [160] use neural networks to represent each word in a way that words with similar contexts are clustered

closer together in the vector space. These techniques generate *static word embeddings* or fixed word embeddings, meaning they represent words as a single vector. However, this method of representing lexical semantics ignores the diversity of word meanings in different contexts, such as polysemy where a single word has multiple meanings or senses. This issue was addressed by introducing *contextualized word embeddings*, which represent words as vectors that vary between contexts when sentences or documents are fed through a pre-trained LM [62] like ELMo [162], BERT [53], and XLNet [220].

Recently, definitions have been used to enhance word vectors, including improving representations of out-of-vocabulary (OOV) terms. Bahdanau et al. [15] train a network to predict words representations based on auxiliary data that describes certain semantic aspects of the word, such as dictionary definitions or linguistic descriptions of named entities sourced from Wikipedia articles. Ruzzetti et al. [179] propose two models to use these definitions to understand the meanings of OOV words: (1) Definition Neural Network (DefiNNet), which uses definition structure to highlight key words, and (2) DefBERT, which uses BERT to convert definitions into a single vector representation. Gajbhiye et al. [76] suggested developing a shared embedding space that combines representations from three strategies: training a concept name (word) embedding model, a mention embedding model since a mention embedding should be similar to the embedding of the corresponding term, and a definition embedding model. Their findings show that the combined embeddings outperform existing strategies in tasks such as ontology completion.

2.3.2 Dictionary Examples and Language Models

Using dictionary examples in NLP applications has not been vastly exploited except for a few exceptions [21, 19]. One reason for this is that some standard LRs actually lack many examples; for instance, 85% of lemmas in WN lack examples. While this presents a challenge for creating connections between LRs and NLP applications,

it also represents an exciting research area. Producing examples from LMs, retrieving them from corpora, or merging them from different resources are all areas with potential for direct and significant impact in NLP. Barba et al. [19] explore a BART-based model for performing the reverse task to DM, i.e., *exemplification modeling*, or generating a dictionary example given a term and its definition. A sequence to sequence architecture is used and trained directly on sense-annotated resources. Also, a transformer-based sequence-to-sequence model by [136] was developed for definition modeling that maps the word's context of usage into its appropriate definition. Considering the background of target readers when generating examples is facilitated by a controllable target-word-aware model [89] that allows users to specify the readability and lexical complexity of generated examples by training the model on discrete control tokens related to these metrics.

Examples Evaluation Evaluating the quality of generated examples is a subjective matter to measure the efficiency of the system and how well these sentences are structured and can deliver the meaning of their words to be used in improving dictionaries for language learners and NLP applications. The evaluation can be done in different ways:

- Intrinsic evaluation (human-made): the sentences can be evaluated by employing human annotators to compare generated sentences with others produced by human lexicographers as in the Generationary model [21]. Also, in the case of Exemplification Modeling [19], human evaluation task was done by asking annotators to measure the fluency of the generated examples which is defined as the rhythm and flow of the language and the sound of word patterns [56], and semantic relationships between words in the generated examples.

- Intrinsic evaluation (automatic): automatic string matching measures are used in automatic intrinsic evaluation like BLEU that is used for machine translation evaluation and compares n-grams matches of the candidate sentence with the ref-

erence sentence[157] and ROUGE which compares an automatically produced summary against a set of reference summaries [121]. However, these metrics are based on simple string matches, and in many cases these are not good indicators of output quality because they score based on a reference sentence and there is no one single good example [21].

- Extrinsic evaluation: extrinsic evaluation is aimed at evaluating the generated sentences based on their impact on the performance of other NLP tasks like WSD. In the Exemplification Modeling, the generated examples are evaluated by using them to train a WSD model [19]. In the Generationary model, the performance of the generated definitions are evaluated by using them in some downstream NLP tasks which are WSD and Word in Context (WiC) to see how they did comparing to the state of the art results [21].

Examples or word usages can serve as a source to obtain contextualized representations of words. The most common approach for distilling word vectors from BERT involves sampling sentences that contain the words of interest. The quality of these sentences influences the effectiveness of the word embeddings. [210] demonstrated various strategies for selecting instances of a given word and found that embeddings trained on clear and informative sentences tend to be of higher quality, capturing more semantic meanings than those chosen randomly.

2.3.3 NLP Tasks

This section introduces Word Similarity and Reverse Dictionary tasks. We focus on these tasks specifically because they form the basis of the analyses in our thesis.

2.3.3.1 Word Similarity

The concept of word similarity involves assigning a metric to a set of items within lists based on the similarity of their meanings [40]. It is important to highlight that *similarity* is different from *relatedness*: semantic similarity involves items that can replace each other in a given context without changing the meaning (e.g., cute and pretty), while relatedness covers a wider range of semantic relations, such as antonymy (e.g., beautiful and ugly), where items are connected but cannot be substituted for one another [177, 31].

Different methods have been proposed to measure similarity and can be classified as word-to-word-based, vector-based, and structural-based [64]. The word-to-word approach measures sentence similarity by comparing individual words. Li et al. [119] applied word-to-word similarity, taking into account semantic information and word order within sentences, with similarity determined using data from a structured lexical database. The vector-based approach represents sentences as vectors that capture the semantic features of words and compares these vectors for similarity. Assessing the semantic similarity between two items reflects the quality of the linguistic representations used, and word embeddings are commonly employed to encode the meaning of words in most NLP applications [174]. The structure-based approach considers sentence structure when calculating similarity, such as the method proposed by Lee et al. [114], which calculates sentence similarity by extracting grammar links, constructing a grammar matrix, and using WN to measure word similarity. Word similarity measures play an important role in different NLP applications such as text mining, question answering, and information retrieval systems. For instance, semantic similarity in information retrieval helps find related documents by understanding word meanings, not just exact matches and this improves the accuracy and relevance of search results [201].

2.3.3.2 Reverse Dictionary

Reverse Dictionary (RD) or concept finder is a helpful application for copywriters, novelists, translators seeking to find words or ideas that might be "on the tip of their tongue" [93]. It is also reflection of the interactions between a speaker and the mental lexicon [228, 229]. This task takes user descriptions or definitions as input and returns words or expressions corresponding to the provided input. RD is a task with a long tradition in lexical semantics, with early methods exploiting hand-crafted rules [22, 187] for extracting textual features. It was Hill et al. [93] who introduced RNNs as suitable architectures that complemented bag of words representations, as well as a dataset specific to RD sourced from different resources. From here, the usage of neural networks first, and more specific, pre-trained transformer encoders like BERT later, have dominated the RD landscape. Among the former, let us highlight, e.g., Pilehvar [166], who integrates WN senses and supersenses as an additional signal, improving over text-based embeddings alone. Further, [224] propose a multi-channel model comprising a sentence encoder based on BiLSTMs and multiple linguistically motivated predictors such as word category (using WN's taxonomy), morpheme or sememe prediction, whereas [39] directly replaces word embeddings with synset embeddings, optionally leveraging examples of usage. An immediate limitation of the above works is their reliance on a sense inventory such as WN, which has proven to work very well for modeling in-domain terminologies, less so for enabling generalization.

Other works have exploited multiple but related tasks in the broad "embedding a dictionary" paradigm, e.g., by combining definition generation and RD with reconstruction tasks via autoencoders [42], or have fine-tuned T5 [168] with excellent results [128]. More recently, LLMs have unsurprisingly been introduced into RD. For example, in a two-stage approach where a fine-tuned LLM first generates a set of candidates which are then passed in a subsequent prompt to a generator for outputting the final set of predictions [197]. Finally, from a more "probing" perspective, RD has been used to gain insights into LLMs representations via conceptual inference, showing that

they encode information about object categories as well as fine-grained features [218].

Despite its usefulness, research in RD is currently limited in two fronts. First, RD benchmarks are mostly sourced from WN and the Oxford Dictionary, and little is known about the effectiveness of RD methods on other resources or languages - with a few notable exceptions such as the multilingual experiments in Yan et al. [219]. This is problematic because the generalization ability of models optimized for these two standard and over-utilized resources might not reflect modern, acquired, rare, evolving or technical terminologies. And second, because there is a surprising lack of work exploiting the generative capabilities of LLMs to improve over embedding-only baselines. While Tian et al. [197] as we mentioned earlier propose to leverage LLMs for RD, their approach requires fine-tuning a text generation model in the first stage, and the final set of predicted terms may not correspond to the vocabulary of the dictionary in question, making this approach hard to apply on large-scale real-world resources[8].

2.4 Summary

This chapter discussed background knowledge related to LRs and briefly described their main components, including definitions, examples, and semantic relations. It also explored how these components are utilized in various NLP tasks. With the information gained from this chapter, we can move forward in the next chapter to discuss the first experiment applied in this thesis that used DM to evaluate WN, the most popular resource in NLP field.

[8] At the time of writing this manuscript, the English Wiktionary has over 7.5M entries (with over 30M entries across all languages), making embedding search a prerequisite on any realistic RD method. `https://en.wikipedia.org/wiki/Wiktionary`

Chapter 3

WordNet under Scrutiny

3.1 Introduction

As discussed in Chapter 1, there is substantial research on evaluating LRs, both intrinsically and extrinsically, with many examples relevant to NLP. We have also highlighted the prominence of WN as a key resource in the NLP and computational lexicography literature. In this chapter, we aim to assess the suitability of WN as an LR, both generally and for specific NLP tasks. Specifically, we present an empirical evaluation of its **dictionary examples**. We argue that despite its widespread use, little to no prior research has examined WN from the perspective of the quality of its examples and their suitability for various NLP tasks. As we will demonstrate, while WN performs well across several metrics and is suitable for many tasks, it also has significant limitations in other applications. This chapter is organized as follows: Section 3.2 provides a closer look at the resources used in this evaluation to compare WN's examples with those of other lexical resources. We then evaluate the examples intrinsically in Section 3.3, by matching them against criteria for *good* dictionary writing and extrinsically in Section 3.4, through definition modeling and word similarity tasks. Finally, the limitations and summary of this work are discussed in Section 3.5.[1].

[1]The code and datasets for the work shown in this chapter are available at `https://github.com/F-Almeman/WordNet_Evaluation`

3.2 Data Resources

We will now discuss the datasets and resources used in the experiments reported in this chapter. **WN** is the primary resource used in this evaluation, and we include two additional resources for comparison:

CHA [38] This is a widely adopted open dataset used in DM, based on data obtained via the OxfordDictionaries.com (2018) API. Each entry consists of a triplet containing the word, its definition, and example(s) showing the word's use in the given sense (31,798 words; average number of examples per definition: 27). The dataset is publicly available online [2]. It is a large and high-quality context-definition dataset, and has been used in different DM works [21, 37]. It was released with two splits, each containing its own train/validation/test sets: *seen*, which tests the pair with (seen word, unseen context, seen definition), and *unseen*, which features a zero-shot test with (unseen word, unseen context) [38]. This is similar to the lexical splits (as opposed to random splits) present in other analogous tasks such as graded lexical entailment [189, 205]. It is worth noting, however, that WN and CHA were built with different objectives. WN was designed to explain how lexical meaning is stored in the mind [29] and is primarily used as a sense inventory [4], while CHA has a different structure. Table 3.1 shows examples from WN and CHA, where it becomes apparent that WN examples have a different pattern, e.g., they are much shorter, and are crucially limited in the contextual information they provide, as opposed to the examples in CHA, which features, first, full-fledged grammatical examples, and second, associated vocabularies that help position the target word in the mental lexicon, which is crucial for word access [229]. Table 3.2 presents detailed statistics comparing the two datasets, offering insight into the resources used in this study. In particular, CHA is substantially larger and contains a higher average number of examples per word. While 12,096 words appear in both

[2] `https://miulab.myds.me:5001/sharing/lWPBRc8hG`

datasets, the overlap in definitions and examples is limited, with only 114 overlapping (word, definition) pairs and a single overlapping (word, definition, example) triplet.

LLM-generated examples For some experiments[3], we expanded the evaluation with dictionary examples generated using ChatGPT (`gpt-3.5-turbo`). We used two different prompts to obtain the examples: the **simple** format (*"Write a sentence showing the word {word}, defined by {definition} in context"*); and the **GDEX** format, which extended the simple prompt with instructions about how the example should be written following GDEX criteria and shown in Appendix A. We leave for future work prompt optimization, which could be done either via careful testing of multiple prompt types (basic, in-context learning, chain of thought, etc.), as well as parametric gradient-based approaches such as DSPY[4].

3.3 Intrinsic Evaluation

There has only been limited work on evaluating the intrinsic quality of LR, specifically WN, by fleshing out their specific features (e.g., type and style of definitions, or readability and informativeness of examples) and studying the extent to which such features dictate the performance of NLP systems. As a proxy for determining the quality of dictionary examples, and given that there is no manually annotated dataset for this purpose, in this evaluation we used the GDEX (Good Dictionary Examples) criteria [108, 20] (see Section 2.2.3.1). The evaluation is conducted in two ways: first through an evaluation that is based on automatic metrics (Section 3.3.1), then complemented by a questionnaire that involves human assessments of examples from several standpoints, namely, naturalness, informativeness, and the extent to which examples are self-contained (Section 3.3.2).

[3]We did not include ChatGPT examples in the automatic evaluation and definition modeling experiment, as they were conducted before December 2022.

[4]`https://github.com/stanfordnlp/dspy`

Source	Lemma	Definition	Example
WordNet	**tall**	Great in vertical dimension; high in stature.	Tall people.
CHA	**tall**	Of great or more than average height especially with reference to an object relative to width.	The elevator came to a stop and the doors slid open revealing the sixth floor of the tall building.
WordNet	**sheet**	Any broad thin expanse or surface.	A sheet of ice.
CHA	**sheet**	A large rectangular piece of cotton or other fabric used on a bed to cover the mattress and as a layer beneath blankets when these are used.	Mary quietly got off the bed and covered him with the sheet and blanket.

Table 3.1: WordNet vs CHA definitions and examples for a given lemma (in bold).

Dataset	# entries	# words	Avg. # ex. / word
WordNet	44,348	20,456	2.17
CHA	785,551	31,798	24.63

Table 3.2: Statistics comparing the WordNet and CHA datasets.

3.3.1 Automatic Evaluation

We based our analysis on the features introduced in GDEX [108] (see Section 2.2.3.1), aiming to implement a systematic approach that closely aligns with the criteria discussed in the original work. Specifically, we chose the following features:

- **Sentence fluency**: This GDEX feature refers to how naturally a sentence reads according to the norms of a language. In computational linguistics, language

models are widely used to estimate the probability of word sequences, supporting tasks such as speech recognition and machine translation by identifying sentences that are more likely to be fluent [108]. Traditional methods assess fluency using frequency statistics or n-gram models, while more recent unsupervised approaches use large language models to evaluate fluency based on perplexity, a measure of a model's uncertainty in predicting words within context [103, 104]. Sentences with lower perplexity scores are considered more fluent, as they align more closely with typical patterns of natural language.

Following these studies, we use the **GPT-2**[5] language model to evaluate how fluent or natural a sentence sounds. This is done by tokenizing the sentence and using GPT-2 to calculate the log-probability of each token given the preceding tokens (next-token prediction). The average log-probability (used to calculate perplexity by exponentiating its negative) is then computed across all tokens in the sentence to estimate how likely the sentence is under the language model. The final fluency score is calculated as follows:

$$\text{Fluency} = \min\left(1, \max\left(0, 1 + \frac{\text{AvgLogProb}}{10}\right)\right) \tag{3.1}$$

Where AvgLogProb is the average log-probability of the tokens in the sentence computed by GPT-2. The division by 10 scales typical GPT-2 log-probabilities (which are around -1 to -10) to a manageable range, and adding 1 shifts the values so that more fluent sentences (with higher average log-probabilities) receive higher scores. The $\min$ and $\max$ operations ensure that the fluency score is bounded within $[0, 1]$, where higher values indicate greater fluency.

For example, the sentence *"She went to the store to buy some milk."* receives a fluency score of 0.72, an average log-probability of -2.75, and a perplexity of

[5]We chose the GPT-2 model because it was state of the art among autoregressive models during our experiments in 2022, while being small enough to run efficiently. Additionally, GPT-2 ensures reproducibility because its weights are fixed, unlike APIs that may be updated in the background.

15.68, indicating high fluency. In comparison, the simpler sentence *"The cat sat on the mat."* yields a fluency score of 0.55, an average log-probability of -4.50, and a perplexity of 90.24, reflecting moderate fluency. In contrast, the meaningless sentence *"The cat telephone banana on mat."* results in a much lower fluency score of 0.06, with an average log-probability of -9.40 and a perplexity of 12,039.

- **Sentence length (len-pen)**: a good example should be between 10 and 25 words long based on Kilgarriff et al. [108]. Accordingly, the length penalty was calculated as zero within the desired length range, and increases as the length moves further away from the target range.

  Let L denote the sentence length (in words), and let the desired length range be $[a, b] = [10, 25]$. We define:

$$d(L) = \begin{cases} 0 & \text{if } a \leq L \leq b, \\ a - L & \text{if } L < a, \\ L - b & \text{if } L > b. \end{cases}$$

$$\text{LengthPenalty}(L) = 1 - \frac{1}{d(L) + 1}.$$

- **Word frequency (freq-pen)**: a sentence was penalized for each non-frequent word, defined as a word which is not among the top 20,000 most common words on the English language, as derived from the Google Web Trillion Word Corpus [28]. This penalty score is derived by dividing the number of non-frequent words by the total number of words in the sentence.

- **Anaphoric references (ana-pen)**: this penalty score was calculated by dividing the number of pronouns in the dictionary example by its total number of words.

Results and Analysis Since Kilgarriff et al. 108 did not specify an optimal weighting for the different factors they took into account in the GDEX metric, we look individu-

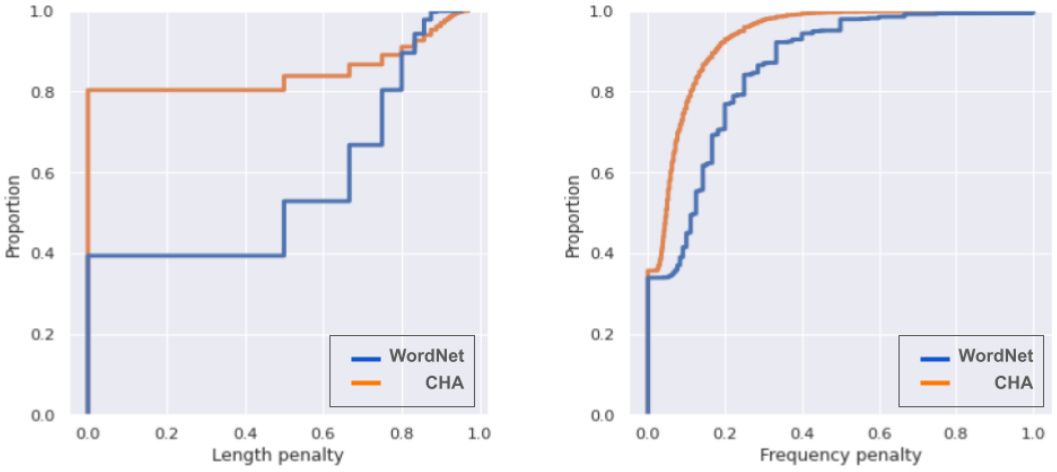


Figure 3.1: Empirical cumulative distribution functions comparing WordNet (blue) and CHA (orange) for length (a) and frequency (b) penalties.

ally at each of these four factors discussed. We leave for future work investigating optimal weighting for these and other metrics, for example, by tuning them on downstream applications. When comparing these scores for both WN and CHA examples, Figure 3.1 (lower is better in both metrics) shows that WN has generally higher penalties both for example length and for usage of infrequent words. Specifically, for instance, we found that 80% of CHA's examples have a length penalty of .6 or less, whereas for the same proportion, the length penalty reaches more than .8 in WN. In a subsequent analysis, we found that these differences, if studied between WN's nouns and verbs, clearly favour nouns, that is, WN's nouns are in general accompanied by better examples. Specifically, we found that, on average, the length penalty is .49 for nouns, and .62 for verbs, and that the frequency penalty is .10 for nouns and .15 for verbs. To illustrate, this WN dataset contains 10,572 noun and 9,823 verb entries, covering 5,593 and 4,359 unique lemmas, respectively. On average, verb lemmas have only slightly more examples (2.25) than noun lemmas (1.89).

While **sentence fluency** is a valid metric, we observe that CHA's examples tend to

	Fluency Score
WordNet	0.3940
CHA	0.4690
WordNet Nouns	0.4810
WordNet Verbs	0.4294

Table 3.3: Average fluency scores across different datasets.

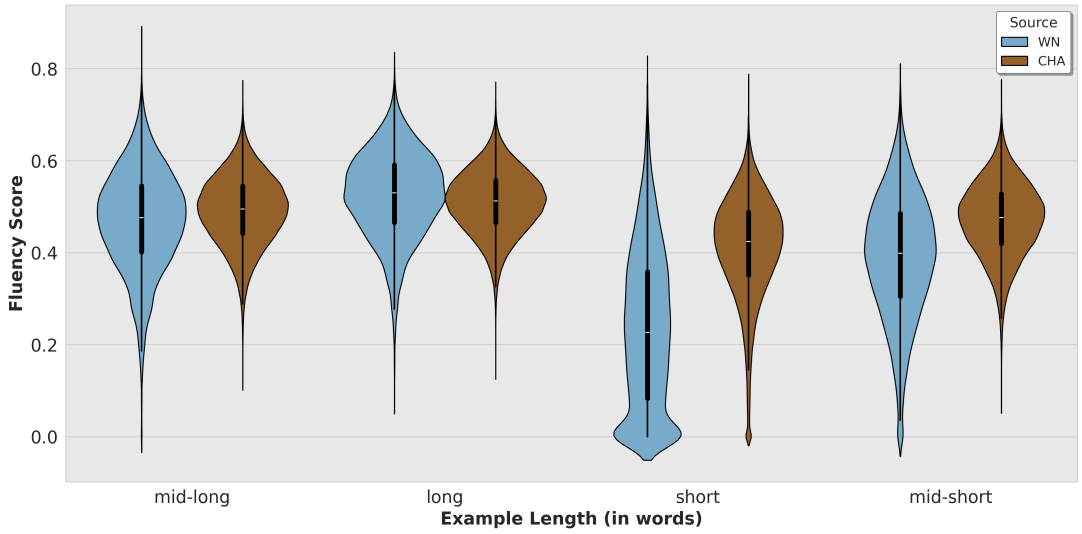


Figure 3.2: Violin plot showing the difference in log-likelihood assigned by GPT-2 to WordNet vs CHA examples (higher is better).

be rated as more fluent on average, as shown in Table 3.3. Notably, although WN nouns and verbs have relatively high scores, the overall WN average is lower due to lower scores for other parts of speech, such as adjectives (0.325) and adverbs (0.425). To explore this further, we split WN's and CHA's examples into four bins as shown in Figure 3.2: *short*, *mid-short*, *mid-long*, and *long*. Instead of fixed length thresholds, we defined these bins using source-specific quartiles based on sentence length measured in words. For CHA, sentence lengths range from 1 to 141 words, with quartiles at 13, 17, and 22 words, resulting in 199,202 short, 192,293 mid-short, 189,754 mid-long, and 171,675 long examples. For WN, sentence lengths range from 1 to 46 words, with

Source	Example	fluency	len-pen	freq-pen	ana-pen
WordNet	How are we going to deal with this problem?	0.79	0.50	0.11	0.22
	He treated his parents thoughtlessly.	0.30	0.83	0.20	0.40
	A full complement	0.46	0.88	0.00	0.00
CHA	This year he has certainly proved his worth and talent ten times over.	0.49	0.00	0.00	0.23
	There are as many recipes for bolognese sauce as there are people who eat bolognese sauce.	0.73	0.00	0.125	0.125
	If your doctor thinks you have a bladder infection he or she will test a sample of your urine to find out if there are bacteria in it.	0.70	0.8	0.00	0.12

Table 3.4: Sample of examples with fluency and penalty scores.

quartiles at 3, 5, and 7 words, resulting in 13,217 short, 10,332 mid-short, 9,722 mid-long, and 11,080 long examples. Examining the fluency scores in these bins, we find that CHA examples tend to be more fluent than WN's in the *short* and *mid-short* bins, while the fluency scores are similar for both sources in the *mid-long* bin. For *long* examples, WN's fluency scores are slightly higher than CHA's, possibly because CHA's sentences in this bin are much longer and thus more challenging to process fluently. Finally, in terms of usage of anaphoric references, we did not find significantly different results between WN and CHA. Table 3.4 shows examples with their evaluation scores.

3.3.2 Human Evaluation

In order to complement the insights derived from the automatic evaluation, we also conducted a comprehensive evaluation of WN examples through a questionnaire. In addition to comparing WN's examples against those from other lexical resources, such

as CHA, we also include examples generated by ChatGPT[6] in our analysis. Our core motivations are to assess the extent to which WN (1) adheres to GDEX criteria; (2) compares with another well-known resource; and (3) compares with examples automatically generated using LLMs. To address these three questions, we recruited three participants who were native English speakers and held higher education qualifications in English or linguistics. This ensured that their responses were informed and reliable for the English dictionary evaluation tasks [182].

Questionnaire Data The questionnaire data was built by including all exact matches words between WN and CHA first, then adding entries with similar definitions for the same term, ranked by cosine similarity, until reaching 400 words [7]. This approach better captures vocabulary overlap and improves sample size and representativeness [55]. Random sampling could include terms with unrelated or inconsistently defined entries, making it difficult to draw reliable conclusions. In contrast, our sampling method improves reliability by ensuring that annotators compare entries referring to the same or closely related concepts, leading to clearer and more meaningful results. The questionnaire focused on these 400 words. Each of these words has a corresponding definition and usage example in both WN and CHA. For each word, we included two generated examples (ChatGPT-simple and ChatGPT-gdex), as described in Section 3.2, resulting in a total of 400 × 4 = 1,600 examples. Each example was evaluated based on three criteria by three annotators, leading to 14,400 unique annotations. Table 3.5 illustrates the kind of examples that participants were asked to annotate. Note that the sources were hidden from the participants to prevent bias and ensure objective evaluations [8].

[6] https://chat.openai.com

[7] We limited the dataset to 400 words to ensure a representative sample while keeping the human annotation workload manageable and within the available budget.

[8] The full questionnaire data is available at the GitHub repository https://github.com/F-Almeman/WordNet_Evaluation/blob/main/datasets/definitions_examples_evaluation_data.pdf

Term	Definition	Example	Source
Cage	Confine in a cage	The animal was caged.	WN
		In future should I leave the house I will cage the dogs no matter who else is in the house with them.	CHA
		The zookeeper had to cage the wild animals to ensure the safety of the visitors.	ChatGPT-simple
		The zookeeper had to cage the wild animal to ensure the safety of visitors.	ChatGPT-GDEX
Ever	At any time	Did you ever smoke?	WN
		They were the cutest couple at our school and no one could ever compare to either of them.	CHA
		Have you ever been to paris?	ChatGPT-simple
		Have you ever been to paris?	ChatGPT-GDEX

Table 3.5: Sample of questionnaire data.

Questions Design The questions presented in the questionnaire were designed to capture participants' perception of the good dictionary examples criteria [12] in a granular way. To this end, the questionnaire was split into two sections: **definitions evaluation** and **examples evaluation**. While our main focus was on the examples evaluation, we included an evaluation of the definitions as well, to assess to what extent the definitions alone were successful in clarifying the meaning of the considered terms.

The primary objective of this exercise is to determine whether (and how much) examples can help readers understand the meaning of difficult or unfamiliar terms that have unclear or difficult definitions. More importantly, we aim to identify if there are important differences in how annotators assess the quality of examples depending on whether they appear with clear or unclear definitions. Specifically, for *definitions evaluation*, participants were presented with a word and its corresponding definition only,

and were asked to assess the extent to which the definition alone clarifies the meaning of the word by assigning one of the following labels:

> **Unclear**: upon reviewing the provided definition, the meaning of the term remains unclear or difficult to comprehend.
>
> **Borderline**: the definition gave me some insight into the term's meaning, but it is still unclear.
>
> **Clear**: the definition clearly and fully explains the meaning of the term.

The *examples evaluation* section of the questionnaire aimed at evaluating the WN and CHA dictionary examples based on GDEX, as well as those generated by GPT-3 and ChatGPT. In this case, annotators were asked to score each example according to the following criteria:

- **Self-containment:** Was the dictionary example fully understandable to you without the need for wider context or consulting external sources? (1–3 scale)
  - 1: No (I had to consult external sources to fully understand it)
  - 2: Partially (I needed some external sources to fully grasp it)
  - 3: Yes
- **Informativeness:** Regardless of your prior knowledge of the term, how well did the example clarify or elaborate on its meaning? (1–5 scale)
  - 1: The example was not informative. Regardless of how much the sentence made sense to me, it did not help clarify the meaning of the term.
  - 5: The example was highly informative; it provided useful and valuable information that helped me gain a better understanding of the term's meaning.

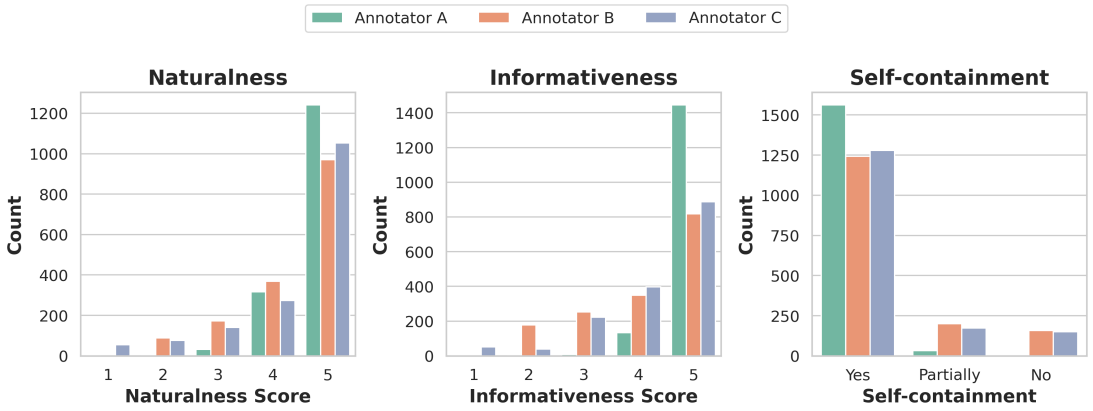


Figure 3.3: Score distributions per annotator for naturalness, informativeness, and self-containment.

- **Naturalness:** How well does this example reflect the style and wording you'd expect to find in everyday language use? (1–5 scale)
  - 1: The example sounds unnatural, synthetic, or awkward.
  - 5: The example uses words and a style that sound natural and that I have encountered or even used myself before.

Results and Analysis Here we present the results of the human evaluation, covering annotator agreement, the assessment of examples using GDEX standards, and the informativeness of examples for difficult definitions. Figure 3.3 summarizes the distribution of annotator ratings across all sources, showing that annotators generally rated examples highly, indicating that they found them to be of good quality.

- **Annotator Agreement** We first measured agreement across the three criteria: naturalness, informativeness, and self-containment. Table 3.6 presents the agreement results using Fleiss' Kappa [69] alongside average pairwise agreement for the three criteria. Fleiss' Kappa measures the degree of agreement among annotators while adjusting for chance, where a value of 1 indicates perfect agreement and 0 corresponds to chance-level agreement [112]. In contrast, average pairwise

	Original		After Binning	
Metric	Fleiss' Kappa	Avg. Agreement	Fleiss' Kappa	Avg. Agreement
Naturalness	0.011	0.516	-0.02	0.78
Informativeness	-0.047	0.454	-0.03	0.71
Self-containment	-0.017	0.733	-0.02	0.74

Table 3.6: Annotator agreement scores before and after converting to 3-category ratings.

agreement directly reports the proportion of cases in which annotators provided the same label, without adjusting for chance.

We observe moderate to high average pairwise agreement across all criteria (0.45–0.73). As shown in Figure 3.3, annotators often distinguished between scores of 4 and 5, so we converted the original 5-point scale into three categories by mapping scores of 4–5 to 3, 3 to 2, and 1–2 to 1, which led to higher agreement scores (0.71–0.78). This suggests that annotators mostly provided the same labels, reflecting strong consistency in their ratings. However, Fleiss' Kappa values remain close to zero, indicating low agreement when adjusting for chance. This difference is due to the imbalance in our data, where most ratings fall into the highest categories. In such cases, even small disagreements can greatly lower the Kappa score despite high raw agreement.[66].

- **GDEX Criteria** Let us now take a closer look at the assessment of the different GDEX criteria for the four considered resources. Figure 3.4 shows the response means and standard errors, which we can interpret as follows. WN examples appear highly natural but are somewhat lacking in informativeness, suggesting they may be easy to understand but not particularly useful. In comparison, the CHA examples are more informative but less natural. Interestingly, GPT-GDEX and GPT-simple lead in informativeness (4.45), indicating that GPT-based examples tend to provide richer, more relevant content. While differences in self-containment are small, GPT-simple slightly outperforms the others (2.80), sug-

gesting it better maintains context within individual examples. Moreover, when comparing `ChatGPT-simple` with `ChatGPT-GDEX`, we cannot see any benefits from the GDEX based prompting strategy. This confirms that, while the GDEX prompt leads to longer and more complex examples, in practice, they prove to be just as effective as the zero-shot approach without explicit instructions.

In order to find out whether the differences in example ratings across sources are statistically significant, we used the Kruskal-Wallis H test [111], a non-parametric alternative to ANOVA that assesses whether ordinal ratings differ across groups without assuming normality. When a significant result was found, we followed up with pairwise Mann-Whitney U tests [129] to identify which specific pairs of sources differed. We applied the Bonferroni correction, which adjusts the significance threshold by dividing it by the number of comparisons to reduce the chance of false positives when running multiple tests. Using these tests, we found significant differences in naturalness (H=14.236, p=0.0026), with pairwise tests revealing that CHA rated significantly lower than those of both WN and `ChatGPT-GDEX` in naturalness ratings. No significant differences were found for informativeness (p=0.26) or self-containment (p=0.64).

In addition, we computed Spearman's rank correlation to examine the relationships between the three example quality metrics. Spearman's correlation measures the strength and direction of a monotonic relationship between two variables based on their ranked values, making it suitable for ordinal data or non-linear associations [192]. The results show significant positive correlations between all pairs of metrics: naturalness and informativeness (0.657), naturalness and self-containment (0.492), and informativeness and self-containment (0.507).

- **Informativeness for Challenging Definitions** We first examine which resource (or pseudo-resource) provided the highest number of informative examples. For this analysis, we specifically focus on words whose definitions received ratings

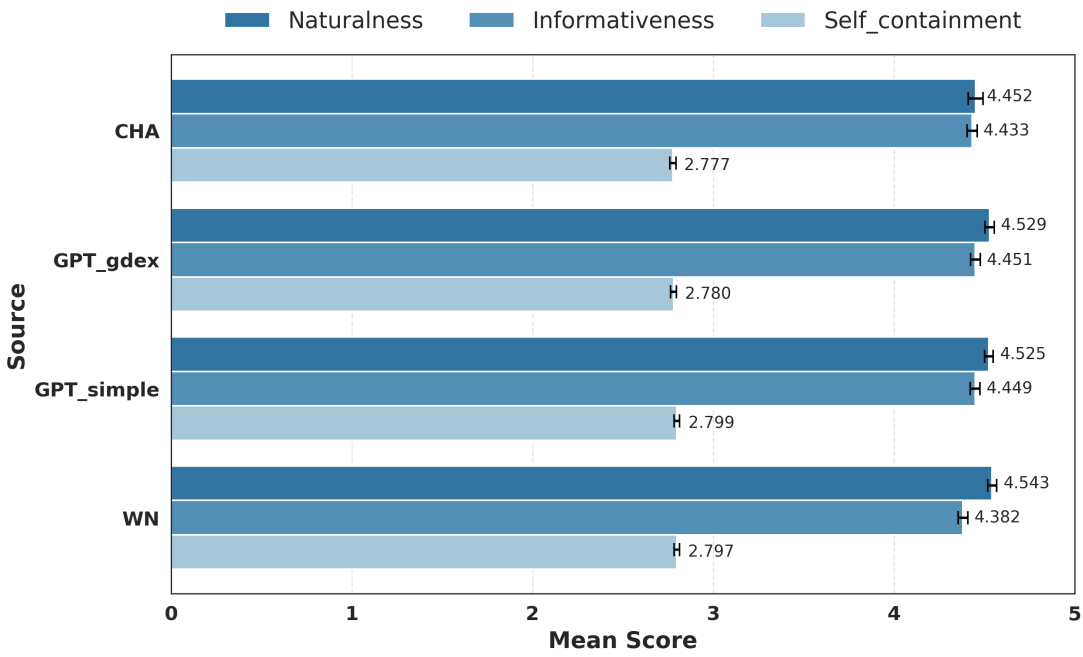


Figure 3.4: Questionnaire results per source

of being *unclear* or *borderline* by at least one annotator. This was the case for 148 definitions. We focus on these 148 words, as the primary purpose of dictionary examples is to help clarify potentially incomplete definitions. This analysis allows us to identify which sources tend to provide more informative examples and which produce less helpful examples when definitions are unclear, highlighting the potential of example augmentation to improve clarity in these cases.

We then examine the informativeness scores of their examples, considering an example informative if it received a score of 4 or 5 from at least one annotator, and uninformative if it received a score of 2 or 1. Finally, we compute the average informativeness scores for each source across examples linked to unclear definitions, separately for the informative and uninformative subsets. As shown in Table 3.7, `ChatGPT-simple` achieves the highest informativeness (4.5) and the lowest uninformative score (2.41), indicating that its examples are generally more informative for definitions needing clarification. In contrast, WN shows the lowest informativeness (4.34) and the highest uninformative score

Source	Informative	Uninformative
WordNet	<u>4.34</u>	<u>2.58</u>
CHA	4.45	2.54
ChatGPT-simple	**4.5**	**2.41**
ChatGPT-GDEX	4.41	2.42

Table 3.7: Average informativeness scores of examples linked to unclear definitions. (bold: best, <u>underlined</u>: worst).

(2.58), suggesting it is less effective in providing clarifying examples for unclear definitions. Additionally, a Kruskal-Wallis H test over this subset revealed that `GPT-simple` performed significantly better than WN in informativeness (U=8921.500, p=0.0048) after Bonferroni correction.

3.4 Extrinsic Evaluation

While intrinsic evaluation explores the intrinsic value of components of WN, namely definitions and examples, we complement it with an extrinsic analysis. In these extrinsic experiments, we aim to determine the impact of WN examples on NLP performance through two tasks: DM and word similarity.

3.4.1 WordNet in Definition Modeling

DM is the task to generate a valid definition for a given input term. This relatively novel task has been approached either with no context (i.e., given a word embedding alone) and, more recently, as word-in-context modeling. Despite their success, most works make little to no distinction between resources and their specific features (e.g., type

and style of definitions, or quality of examples) when used for training. Given the high diversity lexicographic resources exhibit in terms of topic coverage, style and formal structure, it is desirable for downstream definition modeling to better understand which of them are better suited for the task. This section presents an extrinsic evaluation of WN, focusing specifically on its dictionary examples through the task of DM.

3.4.1.1 Experiments

The general formulation of DM is as follows. To generate a definition *d* that defines a target lemma *t* in a context *c*, the standard sequence-to-sequence conditional generation probability is computed by factorising it auto-regressively [21]:

$$P(d|c,t) = \prod_{k=1}^{|t|} P(g_k|d_{0:k-1}, c, t) \tag{3.2}$$

where d_{k} is the k^{th} token of d and d_0 is a special start token [21]. BART, a pre-trained encoder-decoder system, was fine-tuned to perform the definition generation task by taking the pair (context, target lemma) as an input to produce the corresponding definition. The dataset includes (c, t, d) triples where t is the target word (lemma) in a context c (example) and d is the gold definition which defines t in c. The input is encoded as (t, c) pairs and special tokens are used to identify the target lemma in each context such as *The cherry tree <target> bloomed </target>.*, with the lemma "bloom" as the target word in this context.

Exp. 1 (WN vs CHA) Since we are concerned with using WN in definition modeling, we trained and tested the DM model (BART) on WN lemmas that have examples, totaling 44,351 (lemma, definition, example) triples. This number includes repeated lemmas, as a single lemma can be associated with multiple examples. An 80/20 split was used for training and testing. Additionally, we trained the same model using a CHA-derived training set of the same size as our WN training set, and tested it on the

same WN test set. We ensured that no duplicates/leakage occured between sets in both experiments. We train both models with a maximum of 50 epochs with early stopping[9].

Exp. 2 (WN Nouns vs WN Verbs) We trained and tested the same BART model with same hyper-parameters as in the WN vs CHA experiment on random 10k noun lemmas and 10k verb lemmas from WN separately (using again an 80/20 ratio for training and testing) to evaluate whether there are noticeable differences between these two grammatical categories.

3.4.1.2 Results and Analysis

Evaluating the quality of the generated definitions is a subjective matter, as delivering the meaning of words can take many forms. Table 3.8 shows examples of the predicted definitions generated by a WN-trained model and a CHA-trained model. When analyzing these definitions and annotating the error types (following the typification proposed in Noraset et al. [156]), it seems that the predicted definitions generated by the WN-trained model show evidence of under-specificity (first and second rows), since in each case the definition represents the general idea, but where part of the meaning of the target lemma in context is lost. In the third row, the generated definition falls into the self-reference type of error, since it refers to the same lemma in a circular way.

We also noticed that, generally speaking, the CHA-trained model learned to explicitly mention the prototypical concept or the idea to which a definition applies, and this is interesting from a commonsense learning point of view, which has recently received considerable attention [75, 153]. Therefore, given that CHA has many definitions that start with the prototypical concept/entity that embodies that property (e.g., "*accelerate*" having a definition starting with "*of a vehicle*"), for the future, this resource could

[9]We implemented our experiments using the `simpletransformers` (`http://simpletransformers.ai/`) library, a wrapper on top of `transformers` [215].

No.	Lemma	Example	Gold definition	PD_WordNet	PD_CHA
(1)	accelerate	The car accelerated.	move faster	become more powerful or efficient	of a vehicle or aircraft move forward at a high rate of speed
(2)	appear	Did your latest book appear yet?	be issued or published	have a physical form or appearance	of a book or other product reach the shelves of a bookstore or other store
(3)	immigrate	Many people immigrated at the beginning of the 20th century.	come into a new country and change residency	become immigratory	of a person move to a foreign country to settle permanently

Table 3.8: Sample of predicted definitions generated by WordNet-trained model and CHA-trained model. PD_WordNet: predicted definition by WordNet-trained model, PD_CHA: predicted definition by CHA-trained model.

be helpful to map prototypical features to concepts, using dictionary examples as additional contexts.

We evaluated the definitions intrinsically using automatic string matching measures, specifically BLEU, ROUGE-L and METEOR. BLEU is a metric used for machine translation evaluation and compares n-grams matches of the candidate sentence with the reference sentence [157] (we used the default BLEU-4). Rouge-L measures the longest common sub-sequence between the candidate sentence with the reference sentence [121]. METEOR is another improved machine translation evaluation metric that matches uni-grams based on their surface forms, stemmed forms, and meanings [113].

	WordNet	CHA
BLEU	0.18	0.16
METEOR	12.28	14.89
ROUGE-L	16.49	17.37
Gold Def. Len	6.26	6.26
Pred Def. Len	5.49	8.82

Table 3.9: DM evaluation results and definition lengths for WordNet and CHA.

	WN-N	WN-V
BLEU	3.67	0.47
METEOR	20.66	14.13
ROUGE-L	26.85	18.72
Gold Def. Len	9.22	6.48
Pred Def. Len	6.78	4.89

Table 3.10: DM evaluation results and definition lengths for WordNet Nouns vs Verbs.

Exp. 1 (WN vs CHA) Table 3.9 shows the average BLEU, METEOR and ROUGE-L scores for the definitions generated by WN-trained model and CHA-trained model. Overall, the scores for the WN-trained model are generally low, even compared to the CHA-trained model. Additionally, the CHA model produces longer definitions (8.82 words) than the WN model (5.49 words), which may contribute to its higher recall-based scores such as METEOR and ROUGE-L by covering more content from the reference definitions. In contrast, the WN model's shorter outputs may explain its higher BLEU score, which favors precision and penalizes unnecessary or irrelevant words.

Exp. 2 (WN Nouns vs WN Verbs) Finally, with regards to the WN nouns vs WN verbs experiment, Table 3.10 shows the results of the three metrics used for evaluating the generated definitions. When comparing these results and the average of the scores, we can see that the quality of generated definitions of nouns is generally better than that of verbs. Although definition lengths differ, the relative gap is similar, suggesting length is not the main factor. We leave for future work to further explore the differences between WN's noun vs verb examples, and why nouns seem to be easier to learn.

To summarize, the evaluation of WN examples in the DM task indicates that the WN-

trained model often lacks specificity, resulting in circular reference errors and missing key meanings. In contrast, the CHA-trained model successfully includes prototypical concepts, which makes its definitions clearer and more relevant.

3.4.2 WordNet in Word Similarity

For this experiment, we use the examples to generate word embeddings, using MirrorWiC [124], a state-of-the-art model for learning high-quality representations of words or phrases in context. The idea behind this experiment is that informative examples, which were often more contextual, should lead to higher-quality embeddings. To evaluate the quality of the word embeddings, we rely on a number of standard word similarity benchmarks, namely SimLex-999 [92], SimVerb-3500 [80], Stanford's Contextual Word Similarities (SCWS) [95], and MEN Test Collection [30]. We first extracted the common words between WN and CHA along with their examples, and for each word we generated 5 different examples from ChatGPT using the two different prompts (see Section 3.2). Then for each similarity dataset we retrieved the word pairs that can be found in the common words set. For each pair, we computed the cosine similarity between the MirrorWiC embeddings of their associated examples. If a word has multiple examples in WN or CHA, we select the one that leads to the highest similarity score.

	SimLex		**SimVerb**		**SCWS**		**Men**	
	PCC	SCC	PCC	SCC	PCC	SCC	PCC	SCC
WordNet	<u>0.18</u>	<u>0.16</u>	<u>0.21</u>	<u>0.21</u>	<u>0.59</u>	<u>0.54</u>	<u>0.51</u>	<u>0.52</u>
CHA	0.25	0.25	0.28	0.26	0.62	0.58	0.60	0.60
ChatGPT-simple	0.44	**0.43**	0.37	0.36	**0.68**	**0.66**	**0.71**	**0.72**
ChatGPT-GDEX	**0.46**	**0.43**	**0.42**	**0.40**	**0.68**	**0.66**	0.69	0.70

Table 3.11: Correlation between the gold similarity scores and the cosine similarity between examples' encodings (bold: best, <u>underlined</u>: worst).

Results and Analysis By comparing the similarity scores with the gold scores provided by the similarity datasets, we found that ChatGPT examples have the best encoding for all datasets while WN-derived embeddings seem less suitable for the task. This result is consistent with recent findings by Cai et al. [33], who introduced OxfordEval, a metric based on the win rate between generated sentences and Oxford Dictionary, and reported that LLM-generated examples achieved over an 80% win rate. Table 3.11 shows the Pearson's Correlation Coefficient (PCC) and the Spearman's Correlation Coefficient (SCC) between the gold similarity scores and the cosine similarity between examples' encodings.

In addition to the word similarity results, we also list a few illustrative examples [10] (Table 3.12) where, for different word pairs, we show the dictionary examples pair with the highest cosine similarity for each resource. The disparity in the quality of the resources (and GPT generations) becomes apparent. For instance, for *easy* and *tough*, we find that the most similar WN examples are less informative, and most critically, the antonymic relationship between both words is not actually reflected by the given sentence pair. The CHA and GPT generations do not suffer from this issue. A similar situation happens with *dull* and *funny*, where the antonymic relationship is not captured by the WN example pair, and instead we find that both examples elicit health-related senses. CHA, in this case, also falls short (*dull edge* and *funny stomach*), but both of the GPT generated pairs are expressing the sense related to entertainment. Finally, for *rock* and *jazz*, the WN examples pair again shows a conflating of meanings (music, but also exaggerated talk in *"don't give me any of that jazz"*), with CHA and GPT_s both providing accurate music-themed senses. Interestingly, however, GPT_g provides an example pair where a visual arts sense of *jazz* (*"decorated with a jazz theme"*) was found to be most similar to the music sense of *rock*.

[10] The full list of examples are available online `https://docs.google.com/spreadsheets/d/1oWCS2mkw4Fe59XYv1lR1_SIu_LKbEWx6Z1X6B-fRCUA/edit?usp=sharing`

W1	W2	Encoded Example 1	Encoded Example 2	Dict.
easy	tough	An easy victim.	A tough character.	WordNet
		With so many people to choose from, booking several dates in a short amount of time is easy.	Getting published and earning a bit of critical acclaim to spur on further creative efforts is tough for those starting out.	CHA
		She has a natural talent for learning languages, so picking up Spanish was easy for her.	He had a tough time adjusting to his new school.	ChatGPT$_s$
		The hike up the mountain was challenging, but the descent was easy and enjoyable.	The hiking trail was tough, with steep inclines and rocky terrain.	ChatGPT$_g$
dull	funny	Dull pain.	Told the doctor about the funny sensations in her chest.	WordNet
		Most cooks use the point because the edge is dull.	Suddenly my stomach felt funny.	CHA
		His sense of humor was quite dull, and his jokes rarely elicited laughter.	My friend has a funny way of telling stories; he always adds humorous details.	ChatGPT$_s$
		The lecture was so dull that I struggled to stay awake.	The comedian's jokes were so funny that the entire audience couldn't stop laughing.	ChatGPT$_g$
rock	jazz	That mountain is solid rock.	Don't give me any of that jazz.	WordNet
		The movie is a disappointment and could have been a lot better if only he had gone out on a few more limbs than just the inclusion of a few rock tunes.	They're playing a kind of light jazz, something lively to listen to without having to know the words.	CHA
		My favorite genre of music is classic rock.	I love listening to jazz music on a lazy Sunday afternoon.	ChatGPT$_s$
		The concert was held in an open-air amphitheater, and the crowd swayed and danced to the rhythm of the rock music.	The interior of the restaurant was decorated with a jazz theme.	ChatGPT$_g$

Table 3.12: Examples from the word similarity experiment, showing the pair of examples with the maximum cosine similarity between their embeddings (ChatGPT$_s$: ChatGPT-simple and ChatGPT$_g$: ChatGPT-GDEX).

3.5 Summary and Limitations

This chapter discussed the first experiment in this thesis that evaluates WN examples both intrinsically and extrinsically, comparing them with examples from other lexicographic resources as well as content automatically generated by GPT. Our findings highlight that although WN is a valuable resource, particularly suited to a certain type of dictionary example, it is not necessarily the optimal choice when an informative context is required. In downstream evaluation, we trained a sequence-to-sequence definition modeling (DM) architecture based on BART using these examples. The results suggest that WN examples, especially for verbs, may be difficult to learn from, partly due to their short length and stylistic features. Furthermore, in a word similarity task, we found that ChatGPT-generated examples consistently yielded better embeddings across all datasets, indicating a strong preference for generated examples in applications that rely on word or phrase representations.

While these results are promising, several limitations should be acknowledged. First, this chapter focuses primarily on informativeness as the main evaluation metric. While informativeness is important for many NLP tasks, other aspects such as clarity or readability may be more relevant in different contexts and were not explored in detail. For example, in dictionaries aimed at second language learners, clarity and simplicity often take precedence over informativeness. Second, a limitation of the human evaluation is the potential conflation of scores across examples, highlighting the need for future research to develop more refined evaluation frameworks that incorporate a wider range of dictionaries, criteria, definition types, languages, and other relevant factors. Third, the definition modeling experiment introduced in Section 3.4.1 was conducted without exhaustive hyperparameter tuning. We used a BART-based sequence-to-sequence model with moderate fine-tuning, which successfully captured many features of WN examples. However, further improvements could potentially be achieved through more extensive hyperparameter optimization, for instance using LLMs or learned optimizers

to automate selection processes [225, 125]. Finally, although ChatGPT-generated examples outperformed others in embedding-based evaluations across all datasets, these findings are based on a specific experimental setup—particularly the prompts used and the configuration of the ChatGPT model. Additional testing with different models, prompt strategies, or evaluation methods would be necessary to confirm the generality of these results.

Chapter 4

3D-EX : A Unified Dataset of Definitions and Dictionary Examples

4.1 Introduction

Building on insights from previous experiments, which focused on evaluating LRs, and addressing the limitations identified in WN, we explore the task of unifying the landscape of electronic dictionaries around terms, definitions, and dictionary examples. Our hypothesis is that such a resource could greatly benefit lexical semantics and computational lexicography by combining diverse resources. We name this dataset 3D-EX (Dataset of Definitions and Dictionary Examples). This dataset integrates a variety of English dictionaries and encyclopedias into a centralized knowledge repository in the form of <term, definition, example, source> quadruplets. 3D-EX enables exploration of out-of-domain generalization and serves as a unified test bed for tasks in lexical semantics. Table 4.1 shows the motivation behind building 3D-EX: while WN typically provides only one short and often uninformative example, our dataset offers a comprehensive list of examples from diverse resources and dictionaries. In section 4.2, we describe the dictionaries and datasets used to build our dataset. The building process is then explained in detail in Section 4.3, followed by an analysis of the datasets. Next, two experiments have been applied to analyze these datasets in Section 4.4. Lastly,

Section 4.5 outlines the limitations of this work and provides a concluding summary.[1].

Term	Definition	Example
Tiny	Very small	Tiny feet

(a) WordNet

Term	Definition	Example	Source
Tiny	Very small	Tiny feet.	WN
		We can live crowded together in vast cities or as tiny groups in remote deserts.	CHA (Oxford)
		This was to be exchanged after six weeks for the coveted customising tiny diamond.	CHA (Oxford)
		Not long ago, it was difficult to produce photographs of tiny creatures with every part in focus. that's because the lenses that are excellent at magnifying tiny subjects produce a narrow depth of field.	Wiktionary

(b) 3D-EX

Table 4.1: WordNet vs 3D-EX definitions and examples for a given term (in bold).

4.2 Datasets

In this section we review the datasets we integrate into 3D-EX and how they have been applied either in lexicography or downstream NLP tasks. Some resources have been discussed earlier in Section 3.2. The datasets were selected for their unique charac-

[1]The 3D-EX dataset is publicly available online, and the datasets used for building it are also publicly accessible. You can access them through `https://github.com/F-Almeman/3D-EX`

teristics, domain diversity, and relevance to NLP tasks, providing a strong and diverse structure for 3D-EX.

Wikipedia: Wikipedia is an online encyclopedia that has been continuously updated by various contributors since 2001 [48]. We used a dataset that is built by Ishiwatari et al. [99] from Wikipedia and Wikidata and each entry consists of a phrase, description, and example. This dataset is used to evaluate DM approaches that combine distributional and lexical semantics using continuous latent variables [175].

Urban: Urban Dictionary is a crowd-sourced dictionary for terms not usually found in traditional dictionaries. It was created in 1999 and has since become one of the largest databases for non-standard English, with regular updates that reflect current language trends [159]. We used URBAN dataset that was created from Urban dictionary by Reid et al. [175] as a corpus of uncommon and slang words.

Wiktionary: Wiktionary is a freely accessible, web-based dictionary launched in 2002, providing definitions, etymologies, and translations of words. Its content is continuously updated by contributors around the world [16]. It has been used as a resource for WSD [41, 130], especially for retrieving WSD examples which augment labeled data for rare senses [23] and for non-English tasks [91, 183].

Webster's Unabridged : Webster's Unabridged is a version of Webster's dictionary published in 1900 [211], made available through the Project Gutenberg initiative [202]. It focuses on American English and provides detailed word definitions, including background information and usage notes.

Hei++: Hei++ is a dataset that associates human-made definitions with adjective-noun phrases. Since there is no publicly available dataset to evaluate the quality of

definition generation models on free phrases, Hei++ is built by Bevilacqua et al. using the test split of the HeiPLAS dataset [87]. Although Hei++ is a small dataset, it addresses an important gap by focusing on adjective-noun phrases that are not commonly found in traditional dictionaries.

MultiRD: The MultiRD dataset was created by [116] to evaluate a multi-channel RD model that has multiple predictors to predict attributes of target words from given input queries. This dataset uses the English dictionary definition dataset created by Hill et al. [93] as the training set and three test sets: a *seen* definition set, an *unseen* definition set, and a description set that includes pairs of words and human-written descriptions. For each entry, it also includes morphemes, lexical names and sememes.

CODWOE: The CODWOE (Comparing Dictionaries and Word embeddings) SemEval 2022 shared task [137] aimed to compare two types of semantic descriptions, namely dictionary glosses and word embedding representations. This task was applied to multiple languages, and one dataset per language was provided. Each dataset contains a list of examples and, subsequently, each example contains the following key fields: identifier (includes the word), gloss, and embedding-related information.

Sci-definition: Sci-definition is a dataset constructed for the task of generating definitions of scientific terms with controllable complexity [14]. The definitions are drawn from MedQuAD [1] and Wikipedia Science Glossaries[2]. For each term, 10 journal abstracts are provided from S2ORC [127] to allow models to incorporate related scientific knowledge [63, 46].

[2] `https://en.wikipedia.org/wiki/Category:Glossaries_of_science`.

4.3 Building 3D-EX

4.3.1 Data Cleaning

A prerequisite for unifying the above resources into 3D-EX is to perform a number of pre-processing steps to standardize the datasets, remove irrelevant or noisy information, and ensure consistency across all entries. This process includes: lower-casing entries like ophthalmy and OPHTHALMY into a single lowercase form; removing special tokens and any noisy characters such as the `tab` sign; removing entries where their definitions have more than 10% of non alphanumeric characters such as $<$ *Word: maya, Definition: m. i. a. album, Source: Wikipedia* $>$; removing entries that have null values either in words or definitions; removing entries where examples are the same as defined terms such as $<$ *Word: supermaxillary, Definition: supermaxillary, Source: Webster's Unabridged* $>$, and removing duplicate entries within each dataset or split.

While these general steps are applied to all datasets, additional pre-processing was customized for each one based on its specific issues, identified through careful observation and repeated analysis of the data. The main goal was to ensure data consistency and quality, which is detailed in the following steps for each dataset:

Urban: since Urban dictionary contains a high volume of informal language, slang, and non-standard expressions, leading to noisy and sometimes unclear definitions. These definitions are typically very short, with one-word definitions making up 4.33% of entries, some as short as a single character, and many containing a high proportion of emoticons, exclamation marks, and similar features. To handle this, we built a binary classifier based on RoBERTa-base [126], trained on definitions for 2 epochs with two labels to classify definitions as either high-quality or noisy. We used 4,000 positive examples randomly sampled from Wiktionary, CHA, and WN, and 2,000 negative examples from Urban. This classifier, which obtains almost perfect accuracy, is then

applied to the entirety of the Urban dataset, leaving 3D-EX only with Urban entries that are similar to those in more traditional resources, both in content and, more importantly, in style. Table 4.2 lists examples of this filtering process, where we can see Urban-specific properties such as colloquialisms (phrasal verbs, personal pronouns, lack of punctuation marks or high proportion of slang/unknown words).

Term	**Definition**	**Example**	**Filtered**
after john wayne	never name your son after john wayne (or any other famous person)	the gayce's named their son after john wayne	1
pang	pangers pingerz pang pangs pangs MDMA ecstasy	Hi Marissa, it's Frank Recard calling. I'll be in the neighborhood later on, and I was wondering if maybe you wanted to get some pang pangs	1
agyp	pronnounced a-jip	bro went agyp on us, we never thought he would end up with a family and a good job	1
farblegarb	a lot of random garbage	The signal was disrupted, producing a lot of farblegarb	0
citrixify	the process of modifying or altering a computer application for the purpose of publishing the application using Citrix Presentation Server	In order to properly publish that Java-based application, I had to citrixify it so it would run in a seamless window	0
axcellent	when something rocks and is excellent	Dude, that new haircut is axcellent	0

Table 4.2: Examples of Urban entries that were removed vs. retained.

Wiktionary: since some definitions in Wiktionary include etymological information such as when a word was first used (e.g., “first attested in the late 16th century” or “from 16 c”), we removed these parts using regular expressions.

MultiRD: we removed (again, using regular expressions) uninformative definitions such as "see synonyms at" and "often used in the plural".

Sci-definition: in order to construct the **Sci-definition** dataset as <term, definition, example> triples, we took the following steps: from each abstract, we extracted sentences that include the target term, which would act as examples. From these examples, we excluded sentences only containing lists of keywords (typically found in abstracts), and also any example with more than 10% non alphanumeric characters (similarly to our approach to cleaning definitions in Section 4.3.1).

4.3.2 Unification and Splitting

Table 4.3 shows summary statistics for each dataset. Aggregated statistics are provided between two sets, datasets with examples (top) and without (bottom). The last row is related to 3D-EX. It is desirable to keep a reference to the original source for each entry; however, we noticed that there are <`term, definition, example`> duplicates across datasets. For example, the following tuple appears in both WordNet and CHA: `(impenetrable, impossible to understand, impenetrable jargon)`. This is why the final 3D-EX resource contains the SOURCE field as an array containing the sources where that entry was found. Table 4.4 highlights the differences in term, definition, and example lengths across the datasets. Terms are generally short, averaging around one word. Datasets like WordNet and MultiRD provide short definitions, while Sci-definition and Wiktionary provide much longer ones. For example lengths, Sci-definition and Wiktionary contain the longest examples, compared to the shorter examples in WordNet and CHA.

	orig. #entries	**cl. #terms**	**cl. # <T,D>**	**cl. #<T,D,E>**
WordNet	44,351	20,435	36,095	44,241
CHA	785,551	31,798	75,887	752,923
Wikipedia	988,690	162,809	167,569	960,097
Urban	507,638	119,016	145,574	145,896
Wiktionary	145,827	76,453	85,905	140,190
CODWOE	63,596	25,861	45,065	63,137
Sci-definition	8,263	5,281	6,251	166,660
Webster's Unabridged	159,123	89,234	143,782	-
MultiRD	901,200	50,460	671,505	-
Hei++	713	713	713	-
3D-EX		438,956	1,327,342	2,268,225

Table 4.3: Dataset statistics before (orig.) and after (cl.) cleaning, and in terms of unique entries involving terms (T), definitions (D), examples (E).

Furthermore, in terms of splitting 3D-EX for experimentation, it is well known that an issue in word/phrase classification datasets can occur due to a phenomenon known as "lexical memorization" [115], where supervised models tend to associate prototypical features to word types. This has been typically been addressed by releasing two splits, one random, and one known as "the lexical split", where all instances of a given term do not appear across splits [205, 11, 60]. We follow this practice and release 3D-EX with a Random and a Lexical split. Table 4.6 shows examples of entries in 3D-EX and Table 4.5 presents dataset statistics after unification, showing unique instances across both splits. Unique entries are defined as <term, definition, example, source> for datasets with examples (top) and <term, definition, source> for those without examples (bottom).

	Term length			Definition length			Example length		
	min.	max.	avg.	min.	max.	avg.	min.	max.	avg.
WordNet	1	1	1	1	52	7.50	1	46	5.77
CHA	1	1	1	1	71	10.31	2	141	17.86
Wikipedia	1	16	1.84	1	32	6.012	2	40	18.70
Urban	1	31	1.47	1	32	10.01	2	42	11.45
Wiktionary	1	10	1.22	1	100	9.24	2	288	26.52
CODWOE	1	1	1	1	114	10.86	1	214	22.26
Sci-definition	1	11	1.70	2	94	18.49	1	726	25.72
Webster's Unabridged	1	3	1.00	1	90	9.19	-	-	-
MultiRD	1	1	1	1	144	11.72	-	-	-
Hei++	2	2	2	3	23	8.12	-	-	-

Table 4.4: Length statistics per dataset after cleaning (measured in # of words)

	Random split			Lexical split		
	train	**validation**	**test**	**train**	**validation**	**test**
WordNet	26,603	8,788	8,850	27,053	8,573	8,793
CHA	451,191	15,1338	50,394	452,321	157,847	143,949
Wiktionary	84,111	28,127	27,952	89,607	29,176	23,832
Wikipedia	575,554	197,697	186,846	505,964	240,781	213,379
Urban	87,429	29,142	29,325	91,239	29,783	24,881
CODWOE	37,774	12,755	12,608	39,737	12,609	13,166
Sci-definition	101,129	31,766	33,765	106,175	35,966	24,519
Webster's Unabridged	84,802	28,213	28,221	93,423	30,198	19,696
MultiRD	384,295	127,580	128,178	404,114	125,072	112,948
Hei++	426	152	135	428	143	142

Table 4.5: Breakdown of 3D-EX unique entries per split type (random and lexical) and per split.

Term	Definition	Example	source
emergent	coming into existence	an emergent republic	WordNet
word	an (order; a request or instruction); an expression of will	he sent word that we should strike camp before winter	Wiktionary
central london	innermost part of london , england	westminster is an area of central london within the city of westminster , part of the west end , on the north bank of the river thames	Wikipedia
boatie	getting high on a boat	lets go on a boatie this afternoon	Urban
notice	a displayed sheet or placard giving news or information	look out for the notice of the samaritans information evening in the end of september	CHA
worship	to participate in religious ceremonies	we worship at the church down the road	CODWOE
accessory navicular bone	an accessory navicular bone is a small bone located in the middle of the foot	the accessory navicular bone is one of the most common accessory ossicles, which sometimes become symptomatic	Sci-definition
able	having sufficient power, strength, force, skill, means, or resources of any kind to accomplish the object	-	Webster's Unabridged
abbreviation	an abbreviation is a shorter way to write a word or phrase	-	MultiRD
skew picture	an inaccurate or partial representation of a situation	-	Hei++

Table 4.6: Examples of entries available in 3D-EX.

4.3.3 Datasets Analysis

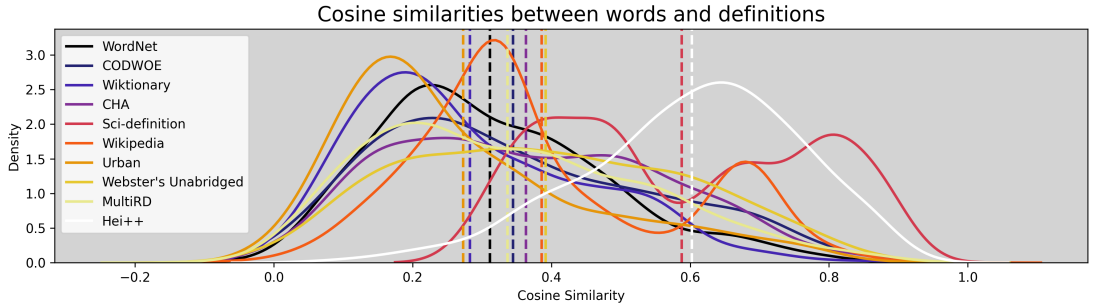
4.3.3.1 Similarity of Dictionary Components in Datasets

To shed some light on how similarities are distributed across datasets, we investigate cosine similarities of their SBERT embeddings [176] using the `all-MiniLM-L6-v2` model, comparing both word-to-definition (WD) and definition-to-example (DE) similaritie (see Figure 4.1). An immediate finding by inspecting these similarities is that `Hei++`, a carefully curated dataset used to evaluate multi-word DM systems, is the one showing the highest similarity between terms and their definitions (Figure 4.1a). This is likely because, first, entries in `Hei++` are specific and do not include generic or frequently used terms. This, along with their rather detailed definitions, contributes to their high similarity scores. On the opposite end of the spectrum, we unsurprisingly find `Urban` dictionary. However, it remains for future work to explore whether its definitions are indeed dissimilar to their corresponding terms, or whether the rarity of the terms leads to low-quality embeddings. Interestingly, we also find that `Sci-definition` exhibits relatively high similarity between terms and definitions.

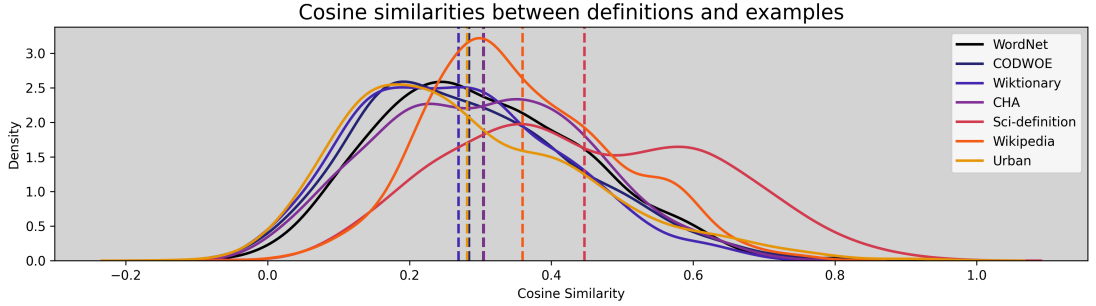
To better illustrate what these cosine similarity scores represent, we present examples that are representative of the *average* similarity in each dataset. For instance, in `Sci-definition`, the term *heat transfer* is defined as "the study of the flow of heat energy; heat transfer concerns dictate major design features of most electrical and electronic systems", (WD similarity: 0.61). Similarly, `Hei++` includes the entry *nagging parent*, defined as "a parent who is constantly scolding his/her children", (WD similarity: 0.60). Meanwhile, `Urban` dictionary includes an entry like *bermuda 84'*, defined as "a female's pubic hair which is shaped in a triangle but long and hairy like back in 1984", (WD similarity: 0.27).

Concerning the cosine similarities between definitions and examples (Figure 4.1b), `Sci-definition` again stands out with higher scores. For example, the term *abyssal plain* is defined as "flat or very gently sloping areas of the deep ocean basin floor", and

used in “the abyssal plain is characterized by low-sedimentation rate thereby being heavily disturbed by bioturbation”, (DE similarity: 0.45). Interestingly, `Wiktionary` shows the lowest overall similarity in this category. For example, the term *global* is defined as “spherical, ball-shaped”, and used in the example “in the center was a small, global mass”, (DE similarity: 0.27). This may suggest that examples in `Wiktionary` are sometimes written to cover broader or more varied contexts than their definitions. As with the case of `Urban` dictionary, a careful semantic analysis of these dictionaries remains for future work.



(a) Word-definition comparison. Hei++ and Sci-definition show the highest average similarities, while Urban shows the lowest.



(b) Definition-example comparison. Sci-definition shows the highest average similarities, while Wiktionary shows the lowest.

Figure 4.1: SBERT-based cosine similarity distributions between (a) word and definition embeddings, and (b) definition and example embeddings across 3D-EX sources. Dashed vertical lines mark the mean similarity for each source.

4.3.3.2 GDEX and Readability-based Examples Evaluation

We assessed the quality of the examples in *3D-EX* using two groups of evaluation metrics. First, we examined the **GDEX** criteria (see Section 3.3.1), which include metrics such as sentence fluency, length penalty (len-pen), frequency penalty (freq-pen), and anaphoric reference penalty (ana-pen), all defined earlier. The main clause (m-clause) GDEX metric is added, where examples with the target word in the main clause are scored 1, and all others are scored 0. To identify the main clause, we used a transition-based dependency parser[3]. In addition to the GDEX criteria, we introduce an additional metric called **Ambiguity**, derived from good dictionary example features outlined by Atkins and Rundell [12], which emphasize the need to avoid gratuitously difficult lexis and potentially confusing lexical items [108]. This concern is relevant to polysemous words, which can confuse learners when multiple senses are possible. Sereno et al. [185] found that readers spent more time and made more regressions when reading sentences with ambiguous words compared to unambiguous ones, showing that ambiguous terms make sentences more difficult to process. Although there is no standard measure for identifying ambiguity in words or sentences [35], we follow their sum-based approach to estimate sentence-level ambiguity by summing the number of WN senses for each word in the sentence, which reflects the overall semantic complexity. For instance, the sentence *"He caught the light bug near the stream"* receives a high ambiguity score of 104 due to polysemous words like *"light"* (not heavy or not dark), *"bug"* (insect or glitch), and *"stream"* (flowing water or data). In contrast, the sentence *"The girl painted a red flower on the wall"* scores much lower at 45, as it consists of more specific and less ambiguous vocabulary. Then, we normalized the ambiguity scores based on sentence length, followed by min-max scaling to obtain values between 0 and 1.

Second, we used readability metrics, which determine the U.S. grade level needed to comprehend a sentence: Flesch–Kincaid Reading Grade Level (FKRGL) [109], which

[3]Implemented with SpaCy: `https://spacy.io/`.

uses word length and sentence length, Dale-Chall Readability (DCR) [51], which is based on sentence length and the number of 'hard' words, and Coleman-Liau Index (CLI) [47], which is calculated using the average number of letters per 100 words and the average sentence length.

Table 4.7 shows the evaluation results, where we report the average for each metric (further statistics such as standard deviation, minimum, and maximum for each metric have been provided in the Appendix B). Our analysis reveals that WN examples exhibit the lowest sentence fluency and often receive higher penalties for short length and the use of ambiguous or multi-sense words. In contrast, WN does well at ensuring that the target word is included in the main clause, likely due to the short length of its examples, and it provides easy-to-read examples as shown by its scores on the readability metrics. In addition, its penalties for using rare words and anaphoric references are low compared to the other resources.

Conversely, Sci-definition examples show the highest fluency scores, suggesting they are more coherent and fluent. Nevertheless, Sci-definition examples demonstrate higher grade levels in all readability metrics, implying a greater level of complexity, which is unsurprising given that they were sourced from scientific journal abstracts. Moreover, Wikipedia has the lowest penalty for sentence length and anaphoric references. This suggests that Wikipedia's examples are closer to the ideal length, use fewer pronouns for clearer communication. Finally, our analysis shows that datasets with higher ambiguity, like Wiktionary and Sci-definition, tend to have higher readability grade levels, indicating more complex text. However, this pattern does not hold for Urban dictionary. This counter-intuitive result might be explained by the large proportion of slang and colloquial lingo. Further analysis could shed light into how to measure readability in Urban Dictionary, considering its obvious idiosyncrasies.

Dataset	fluency ↑	len-pen	freq-pen	ana-pen	ambiguity	m-clause ↑	FKRGL	DCR	CLI
WordNet	0.39	0.71	0.18	0.07	0.59	**0.98**	7.23	8.81	8.23
CHA	0.47	**0.16**	**0.09**	0.09	0.57	0.84	9.12	9.63	9.16
Wiktionary	0.50	0.47	0.23	0.09	**0.54**	0.68	11.81	9.33	10.66
Wikipedia	0.48	**0.16**	0.23	**0.02**	0.55	0.96	11.30	11.30	11.15
Urban	0.41	0.34	0.24	0.12	**0.54**	0.84	**4.42**	**8.62**	**4.92**
CODWOE	0.49	0.51	0.20	0.09	0.55	0.79	9.44	8.77	8.98
Sci-definition	**0.55**	0.35	0.20	0.05	0.57	0.85	17.06	12.26	15.85

Table 4.7: Examples automatic evaluation results (bold: best, underlined: worst, in all metrics, a lower value is better, with the exception of "fluency" and "m-clause").

4.4 Experiments and Results

In order to test the usefulness of 3D-EX, we perform an intrinsic set of experiments where we "stress test" the dataset for artifacts, indirect data leakage (near-synonyms), potential for memorization, etc. This, we argue, is an important step to guarantee 3D-EX can be used for testing lexical semantics models based on it.

4.4.1 Source Classification

In the task of *source classification*, the goal is to, given a <term,definition> instance, predict its original source. We posit that this is an important experiment because it helps us understand how distinct or similar the different sources in our 3D-EX dataset are. In other words, this experiment allows us to identify which sources from our set in 3D-EX are more unique (i.e., easier to classify), and which seem to conflate different lexicographic features (e.g., writing style, coverage or any other artifact). To this end, we fine-tune RoBERTa-base model for three epochs on the training set of 3D-EX. Note that this is a 10-way multi-label classification problem, since for a given

<term,definition> pair, there may be more than one associated source. We report the results of this experiment in Table 4.8, and generally, the lexical split is indeed harder for some datasets, such as WordNet, Urban, and Webster's Unabridged, where we see notable drops in F1 scores. However, for other datasets like Wikipedia and Sci-definition, the performance is nearly identical between the random and lexical splits. In cases like CHA, Wiktionary, and CODWOE, the difference is small, indicating that the lexical split is not substantially harder for those datasets.

	Random Split			**Lexical Split**		
	prec.	rec.	f1	prec.	rec.	f1
WordNet	0.73	0.23	0.35	0.33	0.05	0.09
CHA	0.65	0.48	0.55	0.64	0.47	0.54
Wiktionary	0.80	0.53	0.64	0.65	0.33	0.44
Wikipedia	0.98	0.97	0.98	0.97	0.97	0.97
Urban	0.94	0.87	0.91	0.97	0.66	0.79
CODWOE	0.93	0.55	0.69	0.92	0.42	0.58
Sci-definition	0.99	0.99	0.99	0.99	0.99	0.99
Webster's Unabridged	0.82	0.70	0.76	0.75	0.63	0.68
MultiRD	0.89	0.90	0.89	0.84	0.91	0.88
Hei++	0	0	0	0	0	0
Average	0.77	0.62	0.68	0.71	0.54	0.60

Table 4.8: Source classification results, reported both for the Random and Lexical splits of 3D-EX.

Additionally, Figure 4.2 shows the correlation between F1 and dataset size. When analyzing the effect in the lexical split, it reveals that three datasets (Hei++, WordNet, and MultiRD) exhibit a perfect correlation with performance, indicating that dataset size influences performance for these datasets. Smaller datasets, such as Hei++, tend

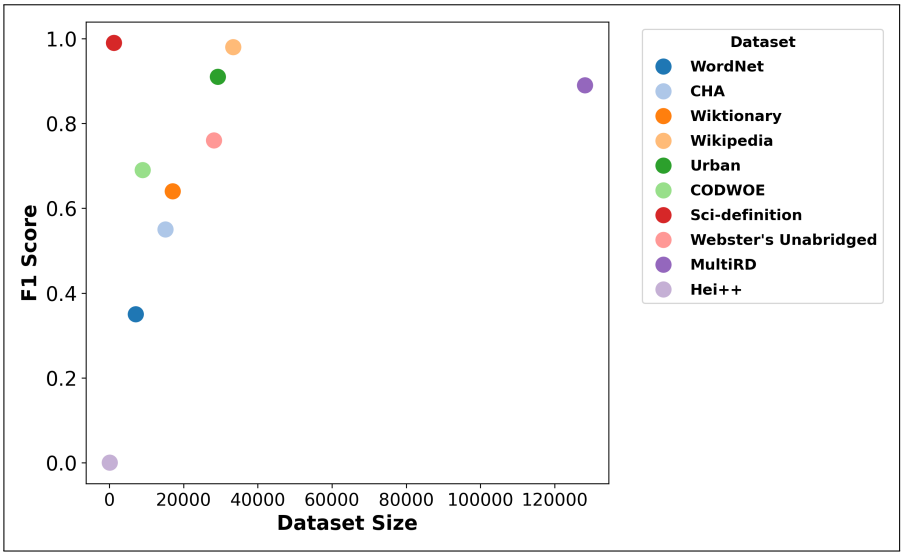
to produce poor performance if they are not sufficiently technical or specific, like Scidefinition, which appears to be an interesting outlier. On the other hand, the largest dataset, MultiRD, demonstrates high performance, indicating that with a sufficiently large dataset, it becomes easier to differentiate between similar resources, such as Wikipedia and Wiktionary. Overall, the correlation between dataset size and performance is moderate, with a Pearson correlation coefficient of approximately 0.4. Overall, it is fair to say that the lexical split is harder on average. Simple averaging across datasets shows lower performance for the lexical split, and this difference is even clearer when using a weighted average based on the number of test examples. The weighted averaged F1 score is approximately 0.79 for the lexical split compared to 0.86 for the random split, indicating the lexical split generally presents a more challenging task.

4.4.2 Reverse Dictionary

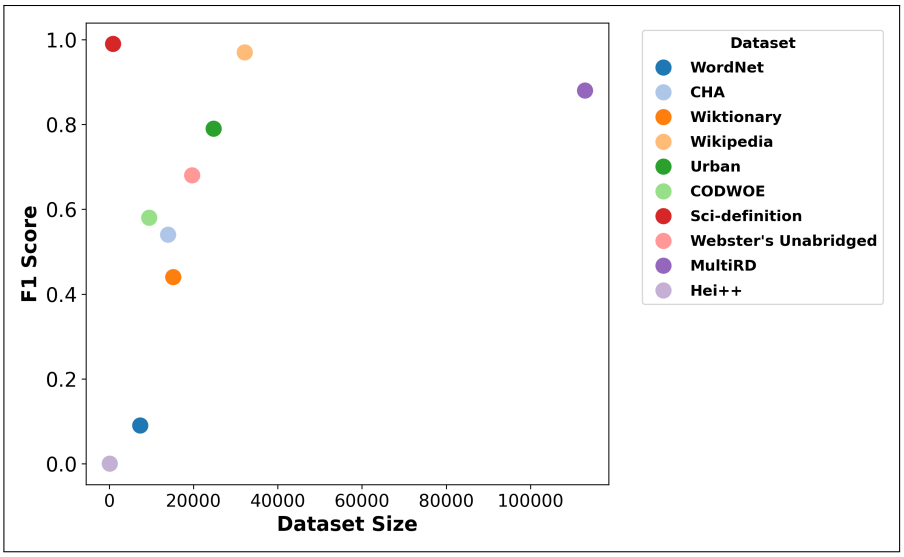
Reverse Dictionary (RD) is a ranking problem in which, given a definition, the task is to retrieve a ranked list of the most relevant words, and it has a long-standing tradition in computational semantics (see more details in Section 2.3.3.2). To establish a set of baseline results on this task, we report results from several embedding models on the test sets of our dataset 3D-EX. Note that while these baselines are unsupervised, we only report results on the test sets (random and lexical) to accommodate future experiments by supervised systems. In terms of evaluation, we report *Mean Reciprocal Rank* (MRR), which rewards the position of the first correct result in a ranked list of outcomes, and its equation is as follows:

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{4.1}$$

where Q is a sample of experiment runs and $rank_i$ refers to the rank position of the *first* relevant outcome for the *i*th run.



(a) Random Split



(b) Lexical Split

Figure 4.2: Correlation between F1 scores and dataset size in source classification for random and lexical splits.

MRR is commonly used in Information Retrieval and Question Answering, but has also shown to be well suited for lexical semantics tasks such as collocation discovery [216, 178].

We use the 3D-EX test set to evaluate the RD task with multiple models. Specifically, we assess the performance of traditional sentence encoding SBERT models, namely `all-MiniLM-L6-v2` , `all-distilroberta-v1` and `all-mpnet-base-v2`. We also evaluate Instructor model [193], an instruction-based encoder that can generate text embeddings tailored to any task given the appropriate prompt. Instructor works by optionally providing the type of the target text (e.g., "a Wikipedia sentence") and the task (e.g., "document retrieval"), to ultimately build a prompt such as "Represent this Wikipedia sentence for retrieving relevant documents". For our use case, we test three variants of Instructor for encoding both words and definitions in our dataset: (1) no instruction provided; (2) providing a generic description of the target text (i.e., "the sentence" for definitions, and "the word" for the terms); and (3) providing a domain-specific description of the target texts (i.e., "the dictionary definition" for definitions and "the dictionary entry" for terms).

We show the results of the SBERT models in Table 4.9, and the Instructor model results in Table 4.10. We can see that even without any instruction prepended to the embedder, the Instructor model outperforms vanilla SBERT models, and that, interestingly, the best results overall in both test splits (random and lexical) are obtained by providing a generic description of target words, and in the random split it is better to not include any instructions for the definitions, while in the lexical split the best performing configuration involves providing detailed instructions for embedding the 3D-EX definitions.

As a final piece of analysis in this section, we apply the same RD approach introduced before to the data from each source in the 3D-EX test set (both random and lexical) individually. We use the Instructor model as it achieves better results than SBERT models with the best-performing configuration: a generic description for target words, no instructions for definitions in the random split, and detailed instructions for definitions in the lexical split. This approach helps us identify which sources are more challenging in this RD task.

Model	Random	Lexical
all-distilroberta-v1	8.41	11.38
all-MiniLM-L6-v2	9.40	13.75
all-mpnet-base-v2	10.98	15.34

Table 4.9: MRR results of the SBERT models.

Random		word		
		no	gen.	dict.
definition	no	14.18	**14.71**	14.56
	gen.	13.64	14.07	14.06
	dict.	14.19	14.59	14.57
Lexical		**word**		
		no	gen.	dict.
definition	no	19.16	20.25	20.02
	gen.	18.70	20.04	19.86
	dict.	19.64	**20.82**	20.60

Table 4.10: MRR results of the Instructor models with different instructions.

From Table 4.11, it can be seen that Wikipedia and Urban dictionaries are the most challenging resources for this task, which could be attributed to either or both dataset size and large number of very similar definitions and terms, as opposed to, for instance, Hei++ or Sci-definition, which are meant to capture unique terms. These are, by nature, more unique when compared to the rest of the lexicon, an insight we revealed when exploring dataset-specifc similarities in Figure 4.1.

Dataset	Random	Lexical
WordNet	32.97	42.27
Wiktionary	50.65	53.05
Wikipedia	9.25	9.19
Urban	18.47	17.49
CODWOE	39.74	46.89
CHA	30.82	35.86
Sci-definition	82.38	82.53
Webster's Unabridged	30.53	34.11
MultiRD	16.69	27.41
Hei++	96.79	94.49

Table 4.11: Breakdown of the RD results in terms of MRR.

4.5 Summary and Limitations

In this chapter, we have introduced 3D-EX, a dataset that unifies different encyclopedias and dictionaries into one single resource. We have conducted an in-depth analysis of the dataset across several splits (random vs. lexical), as well as dictionary source classification and RD experiments. In addition, we included automatic evaluation to assess the examples from different sources within the 3D-EX dataset. Our results suggest that this dataset is both challenging for representation learning methods and promising as a resource for augmenting lexical semantics systems. It has also helped us unveil semantic properties in the different dictionaries and encyclopedias we have integrated into 3D-EX.

While 3D-EX integrates a broad range of lexical resources into a unified dataset, several limitations remain. First, the dataset is English-centric, meaning that its findings may not generalize to other languages without careful adaptation. Second, although diverse sources were included, achieving complete coverage of all available lexical

datasets remains challenging, and additional resources may still offer valuable information. Third, despite the application of cleaning and preprocessing, inconsistencies may remain due to variations in writing style and formatting across sources. Another limitation is the lack of human judgments about the quality or clarity of the entries, which could support more targeted evaluation or learning tasks. Finally, although our current analysis reports results for individual GDEX criteria separately to enable detailed examination, we recognize that GDEX was originally designed to provide a combined score, which we plan to incorporate in future work.

Chapter 5

GEAR: A Simple GENERATE, EMBED, AVERAGE AND RANK Approach for Unsupervised Reverse Dictionary

5.1 Introduction

In the previous chapter, we introduced the RD task through an initial experiment on the 3D−EX dataset, which helped us understand this concept and its challenges. Building on these observations, and as we discussed earlier in Section 2.3.3.2, there has been limited work in expanding benchmarks beyond Hill's dataset [93], the most well-known dataset for RD tasks, especially in terms of different domains and registers. Moreover, no works have explored the seemingly simple generate-then-embed approach, so that many of the practical drawbacks of LLMs (hallucinations and context length limitations, to name a few) could be alleviated, but still using their ability to generate suitable candidate embeddings for KNN search. So, in this chapter, we propose a novel and simple approach to RD that leverages LLMs in combination with embedding models. Despite its simplicity, this approach outperforms supervised baselines in well studied RD datasets, while also showing less over-fitting. In Section 5.2 we describe

the GEAR (**g**enerate, **e**mbed, **a**verage and **r**ank) method and all its steps in details. Then Section 5.3 defines the datasets, and Section 5.4 provides information about our experiments and how we evaluate this method. Our results and analysis are summarized in Section 5.5. In Section 5.6, we examine how different generating models and pooling methods affect the final embeddings. Then, Section 5.7 analyzes the generated terms, specifically their similarities and the impact of similarity on performance. The limitations and summary of the study are presented in Section 5.8. [1].

5.2 The GEAR Method

In this section, we give a brief description of GEAR, a novel, *very* simple, lightweight, and, more importantly, unsupervised method for RD. We denote any dictionary as $D = \{(d_i, T_i)|i = 1, \ldots, N\}$, where d_i is a definition, $T_i = \{t_{i1}, t_{i2}, \ldots, t_{ik_i}\}$ is the set of corresponding terms[2] (i.e., entries in the dictionary), and $k_i \geq 1$ the number of terms associated with d_i. From here, GEAR consists on four simple steps. First, **g**enerate, where, given an input definition d_i, an (LLM) generates a set of possible terms $G = \{g_1, g_2, \ldots, g_m\}$. All throughout this experiment, we use `gpt-4o-mini` [2][3], which we prompt in three different ways to evaluate the effect of varying levels of guidance on the model's performance in generating relevant terms. The intention was to determine whether more detailed prompts, which provide additional context or reasoning steps, would improve the quality of the generated terms.The details for the prompts are in Appendix C, however, at a high level, they are as follows:

- Base prompt 1 (**bp1**): includes a short description of the resource: such as *Given*

[1]The code and datasets for the work shown in this chapter are available at `https://github.com/F-Almeman/GEAR_RD`

[2]It is important to account for a one-to-many relationship at this point because we will be reporting experiments on combinations of multiple resources.

[3]https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

the definition {definition}, generate a ranked list of {k} terms, with the first term being the most related to the definition, assuming they are from the {resource} dictionary. {resource} is {resource description}.

- Base prompt 2 (**bp2**): the same as bp1 but it also includes a sample of terms and definitions from the specified resource to help the model understand the type of terms it should generate such as *These are some examples of definitions and terms in this dictionary: {examples}.*

- Reasoning prompt (**rp**): this final prompt extends bp2 by requesting the generation of examples alongside the terms to explore whether having the LLM 'reason' before answering leads to improved results. Specifically, this part is added: *For each term, provide an example usage in a sentence that matches the style and scope of {dictionary}.*

In the next **e**mbed step, a text encoder $f : \mathcal{V} \rightarrow R^n$ maps each term in G to a vector representation in an n-dimensional space. We use SBERT [176] and the Instructor model [193] to obtain term embeddings and evaluate their performance for comparison (see Section 5.5). SBERT is a traditional and widely accepted model for sentence encoding. Although terms are usually single words, placing both terms and definitions in the same semantic space is not straightforward. Previous work [76] has explored methods that align separate embedding spaces for words and definitions. Instead of using two different models and an extra mapping step, we chose SBERT, which is optimized for sentences but also produces strong embeddings for single words. The Instructor model performed well in the previous chapter, making it a strong baseline for comparison. The resulting matrix $\mathbf{E}_G = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m]^\top \in R^{m \times n}$, where $\mathbf{e}_i = f(g_i)$, is then mean pooled (or **a**veraged) as follows: $\bar{\mathbf{e}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{e}_i$. Finally, in the **r**ank step, given $T = \bigcup_{i=1}^{N} T_i$, which denotes the set of all unique terms in D, we perform KNN search via cosine similarity over T with $\bar{\mathbf{e}}$. The performance of GEAR, just like any other search-based approach, can be evaluated using Information Retrieval metrics that account for different scenarios, e.g., the rank of the first correct term with Mean

Definition	Terms	Sources
Alert and fully informed	[knowing, knowledgeable]	[WN, WN]
River in singapore	[Geylang River, Singapore River]	[WP, WP]
In the middle of the week	[midweek]	[Wik, CHA]
A type of shotgun	[12 gauge, greener]	[Urban, CHA]
An arsonist	[arsonite, torchman, incendiary]	[Wik, Urban, Mul]
The thing which is boasted of	[bragging, brah, brag]	[Mul, Mul, COD]
Perceive an idea or situation mentally	[realized, see, understand]	[Mul, Mul, Mul]
Any supply that is running low	[low supply, short supply]	[Hei++, Hei++]
A term used to describe something so awesome the only way it could be better is if it was between two slices of bread	[ass kicking sandwitch]	[Urban]
An abnormal accumulation of air in the pleural space (the space between the lungs and the chest cavity) that can result in the partial or complete collapse of a lung	[Primary Spontaneous Pneumothorax]	[Sci]

Table 5.1: Examples of `3D-EX` for RD task. Note that one definition could map to more than one term, which in turn can come from different dictionaries. Also note the range of styles and domains. (WordNet (WN), Wikipedia (WP), Wiktionary (Wik), CODWOE (COD), MultiRD (Mul), Sci-definition (Sci)).

Reciprocal Rank (MRR), or the proportion of correct terms at different cutoffs with Precision at k (P@k).

5.3 Data

In this work, we are concerned not only with exploring the usefulness of GEAR when compared to existing baselines, we are also interested in developing an understanding

of what kind of lexical resource poses greater challenges to this method as well as its components alone. For this reason, we perform experiments on two distinct but complementary datasets.

As a first evaluation set, we use the three test sets from the Hill et al. 93 dataset (Section 5.4.1) to compare GEAR with published RD methods: the seen set, which includes 500 word-definition pairs from the training set to evaluate recall; the unseen set, containing 500 pairs where both the words and definitions are excluded from training; and the description set, consisting of 200 words with with human-written descriptions. Both the *unseen* and *human description* datasets are suitable for determining the generalization ability of any tested method. Secondly, we report performance on the dictionaries included in our `3D-EX` dataset (Section 5.4.2), a comprehensive resource that integrates multiple dictionaries and organizes them into `<word,definition>` pairs and `<word,definition,example>` triplets[4]. We convert this dataset into a suitable RD format, namely `<definition, list of terms>` and perform two types of splits: a *random split* and a *source split*. In the random split, the data is split randomly into 60% for training, 20% for validation, and 20% for testing. In the source split, a stratified split is used, where definitions from each source in the dataset are extracted into separate datasets, and then each dataset is split while maintaining the same 60%, 20%, and 20% ratio for training, validation, and test sets. Despite the unsupervised nature of our work, we still conduct all our experiments on the test splits alone to enable comparison in further iterations with supervised methods. Table 5.1 shows examples of entries in `3D-EX`, and illustrates the diversity in register, domain and style within the resource.

Model/Method	Seen Definition			Unseen Definition			Description		
	mr	acc@k	rv	mr	acc@k	rv	mr	acc@k	rv
OneLook	**0**	66/.94/.95	200	-	-	-	5.5	.33/.54/.76	332
BOW	172	.03/.16/.43	414	248	.03/.13/.39	424	22	.13/.41/.69	308
RNN	134	.03/.16/.44	375	171	.03/.15/.42	404	17	.14/.40/.73	274
RDWECI	121	.06/.20/.44	420	170	.05/.19/.43	420	16	.14/.41/.74	306
SuperSense	378	.03/.15/.36	462	465	.02/.11/.31	454	115	.03/.15/.47	396
MS-LSTM	**0**	**.92/.98/.99**	**65**	276	.03/.14/.37	426	1000	.01/.04/.18	404
Multi-channel	16	.20/.44/.71	310	54	.09/.29/.58	358	2	.32/.64/.88	203
BERT	**0**	.57/.86/.92	240	18	.20/.46/.64	418	1	.36/.77/.94	94
RoBERTa	**0**	.57/.84/.92	228	37	.10/.36/.60	405	1	.43/.85/.96	46
GEAR_bp1	**0**	.66/.84/.96	200.122	**0**	**.70/.88/.97**	**180.955**	**0**	.89/.99/.99	70.5334
GEAR_bp2	**0**	.71/.88/.97	170.451	**0**	.65/.82/.95	225.324	**0**	**.93/.99/1**	**1.57314**
GEAR_rp	**0**	.70/.87/.96	185.97	**0**	.66/.86/.96	190.8	**0**	.91/.99/1	1.7837

Table 5.2: GEAR results on the Hill dataset compared to competitor models (using the Instructor model for embeddings, as it achieves the best performance), according to median rank (mr), accuracy@k (acc @ 1/100/1000), and rank variance (rv). Baselines results are from Zhang et al. [224] and Yan et al. [219].

5.4 Experiments

5.4.1 GEAR on Hill's Dataset

We introduce two sets of experiments on Hill's dataset. The first uses an LLM alone to perform RD by directly generating terms from input definitions. The second integrates the LLM with embedding models to form GEAR, which enhances performance across

[4]While dictionary examples are a valuable resource for improving text representations, we leave them out of our experiments in order to limit the number of components to test.

the board. For evaluation, we compare GEAR results with the following baselines: (1) OneLook, which is the most popular commercial RD system [224]; (2) BOW and RNN with rank loss [93], which are neural models where BOW uses a bag-of-words approach and RNN employs Long Short-Term Memory (LSTM); (3) RDWECI [143], which improves BOW by adding category inference; (4) SuperSense [166], which advances BOW by using pre-trained sense embeddings; (5) MS-LSTM [105], which enhances RNN with WordNet synset embeddings and a multi-sense LSTM; (6) Multi-channel [224]; and (7) BERT and RoBERTa [219], which are trained to generate the target word for the RD task. We use three evaluation metrics based on previous work: median rank of target words (lower is better), accuracy of target words in the top K results (higher is better), and rank variance (lower is better). Table 5.2 shows that GEAR outperforms all baselines on both the unseen definition set and the description set. We can also observe that MS-LSTM performs effectively on the seen definition set but not on the description set, showing its limited ability to generalize [224].

5.4.2 GEAR on 3D-EX

Despite the new SoTa established on Hill's dataset, we are also interested in exploring key components, such as LLMs and embeddings, applied to other resources. To this end, using 3D-EX as a test bed, we apply the same set of experiments described in Section 5.4.1 and introduce a new experiment that uses only different text embeddings, excluding the **g**enerate step. This additional experiment will serve as a baseline for understanding the performance of these simpler approaches, and it will help assess the extent to which GEAR can provide a significant improvement over these basic and untuned approaches.

In terms of experimental setup, and unless otherwise specified, we consistently evaluate a fixed-length ranked list of 5 terms for each given definition, where each term in the list is ranked according to its relevance to the definition. These terms are then compared to the corresponding gold terms to assess the quality and relevance of the

results. Regarding the evaluation metrics, we employ *Mean Reciprocal Rank* (MRR), which rewards the position of the first correct result in a ranked list of outcomes (see Equation 4.1). Additionally, we use *Precision @ k* (P@k), which calculates the precision of relevant items within the top k positions of a ranked list, and is defined as follows:

$$\text{Precision@k} = \frac{1}{k} \sum_{i=1}^{k} \text{rel}_i \tag{5.1}$$

where rel_i is 1 if the item at position i is relevant and 0 otherwise.

Embeddings We evaluate the performance of different embedding models, which we select considering different factors such as adoption among the community, performance in open benchmarks such as MTEB [145], availability in the HuggingFace[5] hub, as well as being of manageable size. These models are as follows.

- Sentence Transformers (SBERT) [176] models, specifically focusing on the following three variants:`all-MiniLM-L6-v2`, `all-distilroberta-v1`, and `all-mpnet-base-v2`.
- Jina Embeddings [85], which is a language model trained on Jina AI's Linnaeus-Clean dataset, containing query-document pairs obtained from different domains. We use these two versions: `jina-embedding-b-en-v1` and `jina-embedding-l-en-v1`.
- General Text Embeddings (GTE) model [120], which is trained on a large-scale corpus of relevant text pairs from different domains, allowing the GTE models to be used in a range of downstream text embedding tasks. In this work we use `gte-large`.

[5] `https://huggingface.co/`

- Instructor [193] model, which generates text embeddings for different tasks (such as classification or retrieval) and domains (such as science or finance) based on task instructions. We use in this work `instructor-large`. Furthermore, we examine three different variants of instructions for encoding terms and definitions as previously specified in Section 4.4.2.

- Universal AnglE Embedding [118], another instruction-based encoder, and which we use with the same configurations as Instructor. We use `UAE-Large-V1`.

5.5 Results and Analysis

5.5.1 Hill's Dataset

Table 5.3 shows how the performance improves with the GEAR method. In the first part of the table, where candidates are evaluated without any embeddings, the likelihood of having the target term in the top 5 candidates is low. Among the models tested with GEAR, the Instructor model, which encodes terms as dictionary entries based on instructions, performs best, which gains of above 10% on the seen and unseen splits, but negligible differences on the human descriptions (presumably because these already get a good sentence embedding from sentence bert, on one hand, and also because they might not be accurately described as dictionary resources, which is what we used as an instruction for Instructor). For prompt effectiveness, we found that adding the requirement to generate a dictionary example, and despite its usefulness in other settings, does not improve over base prompt 2, which simply provides as input a few exemplars.

Embed. Model	Split	Prompt	ACC@1	ACC@5
No embeddings	S.	bp1	26.8	44.0
		bp2	30.4	50.2
		rp	29.2	46.8
	U.	bp1	30.0	47.2
		bp2	33.4	53.8
		rp	33.4	49.8
	D.	bp1	70.0	77.0
		bp2	72.5	83.0
		rp	72.0	81.5
SBERT	S.	bp1	57.8	68.2
		bp2	62.4	73.0
		rp	60.6	72.2
	U.	bp1	59.0	70.8
		bp2	63.8	77.0
		rp	61.6	75.2
	D.	bp1	90.5	96.5
		bp2	93.5	97.5
		rp	94.0	98.0
Instructor	S.	bp1	66.0	80.6
		bp2	71.4	84.6
		rp	70.4	84.0
	U.	bp1	64.6	79.0
		bp2	70.0	83.4
		rp	66.4	82.4
	D.	bp1	89.5	98.0
		bp2	92.5	98.5
		rp	91.5	99.0

Table 5.3: Performance comparison of LLMs (no embeddings models) and GEAR. S.: Seen split, U.: Unseen split, and D.: human description split. Prompts are bp1 (Base Prompt 1), bp2 (Base Prompt 2), and rp (Reasoning Prompt).

5.5.2 3D-EX Dataset

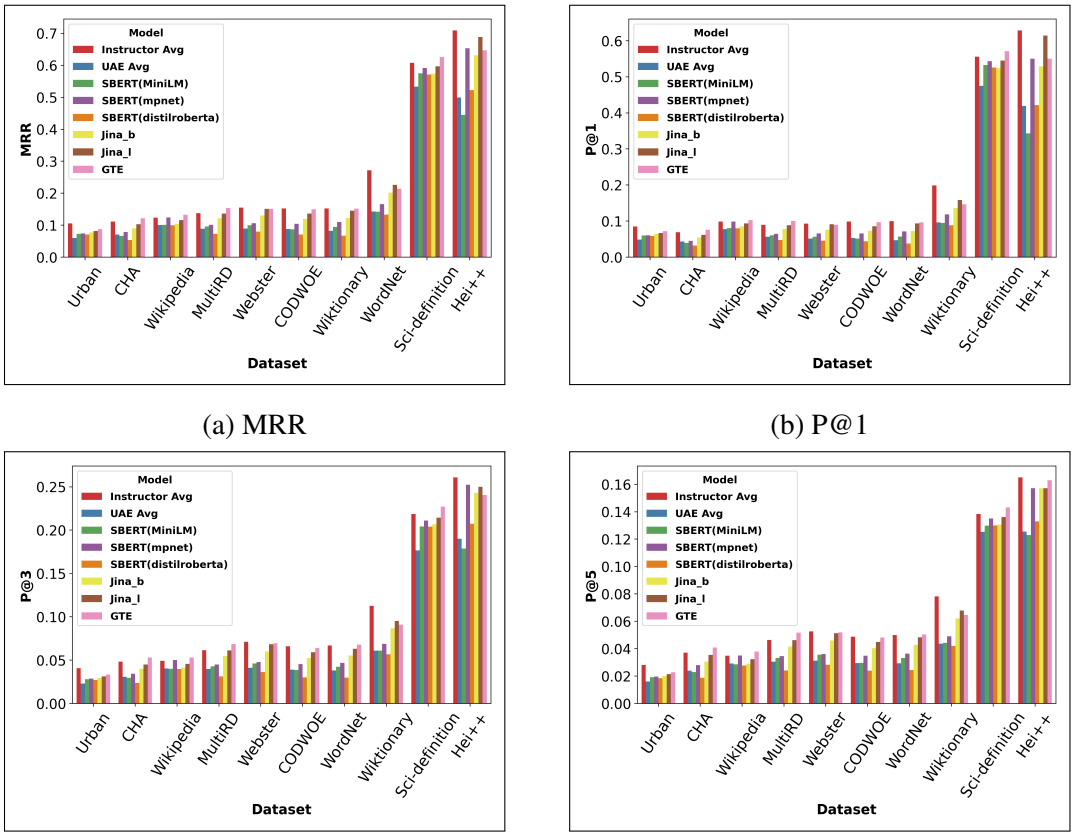
Table 5.4 presents the average MRR and P@1/3/5 for each prompt across various resources in 3D-EX, comparing two methods: one without embedding models and the other using the GEAR method. As previously demonstrated in Hill's results (Section 5.5.1), the GEAR method, particularly with the Instructor model, achieves the best performance, showing improvements in MRR and P@1 ranging from 3 to 6 points. Concerning the type of prompt, interestingly, we found the sophistication of the prompt to matter the most when combined with embedding models (where we see an improvement of around 7% MRR from base to reasoning), but only 2% when prompting alone is considered.

Model	Prompt	MRR	P@1	P@3	P@5
No embeddings	bp1	28.27	24.58	10.64	6.91
	bp2	30.21	26.18	11.39	7.42
	rp	30.99	26.98	11.70	7.58
SBERT	bp1	40.61	33.75	17.36	11.96
	bp2	43.01	36.21	18.32	12.59
	rp	44.21	37.09	18.92	12.99
Instructor	bp1	43.47	36.41	18.00	12.24
	bp2	45.58	38.67	18.76	12.73
	rp	46.37	39.31	19.09	12.98

Table 5.4: Performance comparison of LLMs (no embeddings models) and GEAR methods across different models and prompts in 3D-EX, showing the average score across different dictionaries.

Table 5.5 shows the average MRR and P@1/3/5 for each embedding model mentioned in Section 5.4.2. Results are much lower compared to those achieved with the GEAR method, as well as below prompting alone. In Figure 5.1, we illustrate the performance

across these different datasets, and verify that Hei++ and Sci-definition have higher values, while Urban and CHA show lower values. This variation is likely due to the nature of the entries in Hei++ and Sci-definition, designed to capture more specialized and unique terms. We see, interestingly, that while Instructor embeddings alone are consistently outperforming the rest, they particularly shine in WordNet, which suggest that WordNet embeddings may benefit from additional context to the encoder, since it has been shown that WordNet's definitions and examples are perhaps too short to be informative [81].



(a) MRR

(b) P@1

(c) P@3

(d) P@5

Figure 5.1: Performance comparison for various embedding models across different metrics in 3D−EX.

Model	MRR	P@1	P@3	P@5
Instructor (dict. - dict.)	24.88	19.80	9.82	6.73
Instructor (gen. - dict.)	24.81	19.40	9.96	6.78
Instructor (gen. - no)	24.73	19.72	9.78	6.68
Instructor (dict. - no)	24.51	19.55	9.68	6.60
Instructor (gen. - gen.)	24.32	19.17	9.68	6.65
Instructor (dict. - gen.)	24.15	19.05	9.55	6.61
Instructor (no - no)	24.03	19.23	9.45	6.47
Instructor (no - dict.)	23.92	18.91	9.51	6.50
Jina (large)	23.78	18.96	9.33	6.40
GTE (large)	23.47	18.18	9.38	6.56
Instructor (no - gen.)	23.24	18.31	9.20	6.37
UAE (gen. - gen.)	22.93	17.63	9.25	6.41
UAE (dict. - gen.)	21.99	16.96	8.77	6.16
UAE (gen. - dict.)	21.79	16.80	8.71	6.12
UAE (dict. - dict.)	20.90	15.87	8.45	5.94
Jina (base)	20.85	16.11	8.38	5.81
all-mpnet-base-v2	20.14	15.96	7.96	5.45
UAE (gen. - no)	18.91	14.48	7.66	5.36
UAE (dict. - no)	17.77	13.55	7.20	5.01
all-MiniLM-L6-v2	17.04	13.07	6.84	4.82
all-distilroberta-v1	16.55	13.05	6.54	4.50
UAE (no - gen.)	8.98	7.25	3.51	2.39
UAE (no - dict.)	8.10	6.50	3.16	2.15
UAE (no - no)	5.34	3.81	2.24	1.58

Table 5.5: Comparing different embedding models without any support from an LLM-based generation step, showing the average score across `3D-EX` dictionaries.

5.6 GEAR Components Analysis

Generation with Open Source LLM To ensure that our approach is both effective across different models and accessible for future research, we repeated the GEAR experiment on the Hill dataset using Llama [199], specifically `Llama 3.1-70B` [6], an open source and freely available model. Table 5.6 presents the results compared to using `gpt-4o-mini`. In comparison to the competing systems presented in Table 5.2, our method consistently outperforms them on 2 of the 3 datasets. This suggests that the specific model used for generation is less critical than ensuring the generated terms are semantically meaningful and lead to a reliable representation for retrieval.

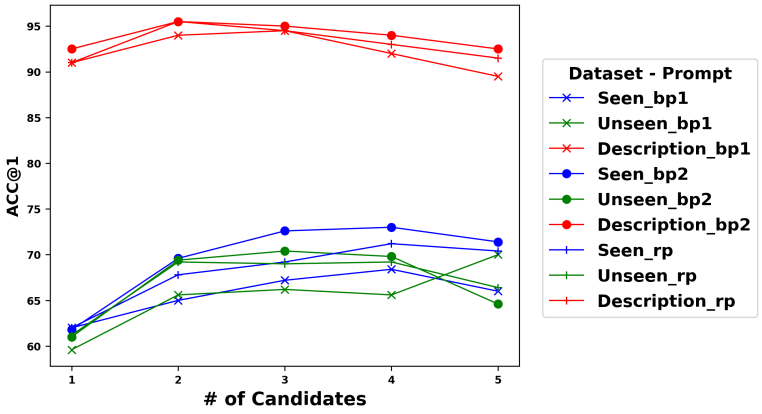
Different Pooling Methods In order to understand the effect of the number of candidates produced in the generation step, we plot performance at different values for all three splits and for accuracy@k. Figure 5.2 shows that using just one candidate is not optimal, while averaging over 2 or 3 candidates provides better results, sometimes outperforming all the 5. These results also suggest that while we could have tuned the candidate number on a development set, with the tools we tested (`gpt-4o-mini`; and Instructor and SBERT), it seems proven that performance plateaus at only a handful of generated terms.

Additionally, We explored the effectiveness of max pooling instead of averaging the generated term embeddings. These experiments did not provide any improvements over the averaging method results. As we can see in Table 5.7, for the full GEAR method using **bp1**, the results were around 1-2% worse for all 3 ks in accuracy@k. Similarly, for the other two prompts, we found a consistent underperformance when compared with averaging, again between 1% and 2% below, with the performance on Hill's test set going further below, up to 4%.

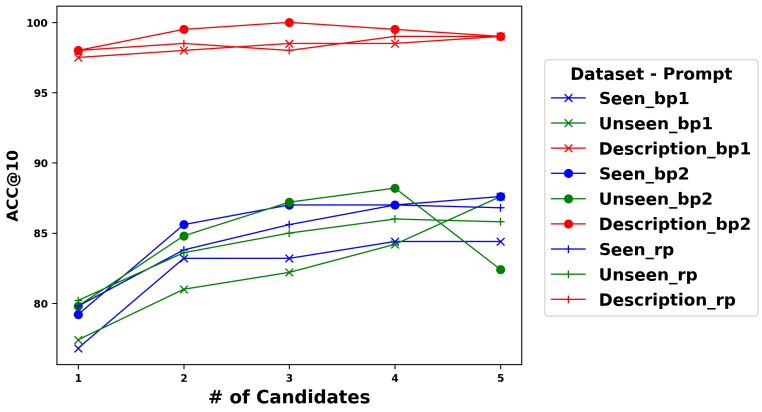
[6] `https://huggingface.co/meta-llama/Llama-3.1-70B`

Split	Prompt	ACC@1	ACC@10	ACC@100
LLaMA-based Candidate Generation				
S.	bp1	55.6	67.4	76.8
	bp2	69.8	83.6	94.2
	rp	76.2	89.0	97.0
U.	bp1	68.8	84.4	92.4
	bp2	67.2	85.0	93.2
	rp	69.8	87.8	95.2
D.	bp1	88.0	97.5	99.5
	bp2	88.5	98.5	99.5
	rp	91.5	100.0	100.0
gpt-4o-mini-based Candidate Generation				
S.	bp1	66	84	96
	bp2	71	88	97
	rp	70	87	96
U.	bp1	70	88	97
	bp2	65	82	95
	rp	66	86	96
D.	bp1	89	99	99
	bp2	93	99	100
	rp	91	99	100

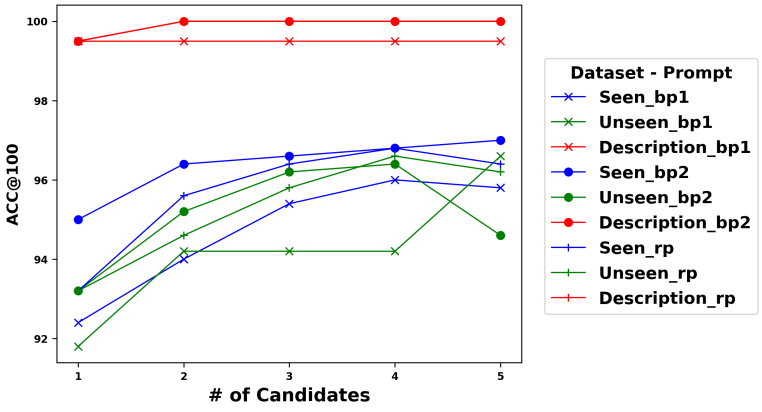
Table 5.6: GEAR results using LLaMA and gpt-4o-mini for candidate generation, with Instructor for embeddings. Evaluated on Hill's dataset across different prompts and data splits. S.: Seen, U.: Unseen, D.: Description. Prompts are bp1 (Base Prompt 1), bp2 (Base Prompt 2), and rp (Reasoning Prompt).



(a) Accuracy@1



(b) Accuracy@10



(c) Accuracy@100

Figure 5.2: A comparison of the performance on Hill's splits, evaluating the number of candidates in the generate step, which are then subsequently averaged to produce the input vector for KNN search.

Split	Prompt	ACC@1	ACC@10	ACC@100
		Max Pooling		
S.	bp1	63.6	81.2	94.0
	bp2	69.0	84.6	95.8
	rp	65.6	83.4	94.6
U.	bp1	61.6	80.4	93.8
	bp2	68.2	86.2	95.0
	rp	66.6	85.2	95.2
D.	bp1	90.0	97.5	100
	bp2	90.5	97.0	100
	rp	90.5	98.0	100
		Average Pooling		
S.	bp1	66	84	96
	bp2	71	88	97
	rp	70	87	96
U.	bp1	70	88	97
	bp2	65	82	95
	rp	66	86	96
D.	bp1	89	99	99
	bp2	93	99	100
	rp	91	99	100

Table 5.7: Comparison of max and average pooling results across different prompts and splits in Hill's dataset using the Instructor model for embeddings. S.: Seen, U.: Unseen, D.: Description. Prompts are bp1 (Base Prompt 1), bp2 (Base Prompt 2), and rp (Reasoning Prompt).

5.7 Generated Terms Analysis

In this section, we focus on analyzing the terms generated by LLMs in GEAR, with a particular emphasis on examining their similarities to gain insight into their diversity and how they influence GEAR performance. Appendix D shows samples of the generated terms.

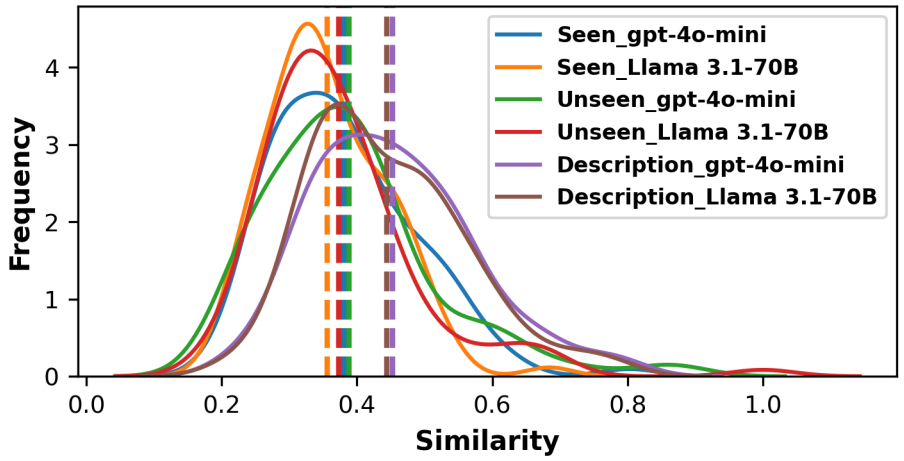
To analyze the semantic similarity among the five predicted terms generated by the LLMs for each definition, we calculated pairwise cosine similarities using SBERT. For each set of candidates, we first encoded the five terms using the SBERT model, specifically `all-MiniLM-L6-v2`, to obtain their embeddings. We then calculated cosine similarities between all unique pairs, forming a 5×5 similarity matrix. From the similarity matrix, we extracted the upper triangular values (excluding the diagonal), corresponding to the 10 unique pairwise similarities among the 5 candidates. We then averaged these values to obtain a single similarity score for each set of candidates. This process was applied to 100 randomly sampled definitions from the Hill's and 3D-EX datasets, using the best-performing prompts — *Base Prompt 2* for Hill's and *Reasoning Prompt* for 3D-EX. The results of this analysis are shown in Figure 5.3.

The first plot in the figure compares the similarity scores between candidates generated by the two models, `gpt-4o-mini` and `Llama 3.1`, for the Hill's datasets (seen, unseen, and description) using *Base Prompt 2*, which achieves the best results in terms of performance. Overall, `gpt-4o-mini` outperforms `Llama 3.1` across all datasets. For the seen dataset, `gpt-4o-mini` shows higher and more consistent similarity scores for the generated terms compared to `Llama 3.1`. In the unseen dataset, `gpt-4o-mini` shows a similar pattern with consistent scores, while `Llama 3.1` displays a wider range but lower scores. In the description dataset, `gpt-4o-mini` achieves its highest similarities, whereas `Llama 3.1` shows lower scores with some outliers.

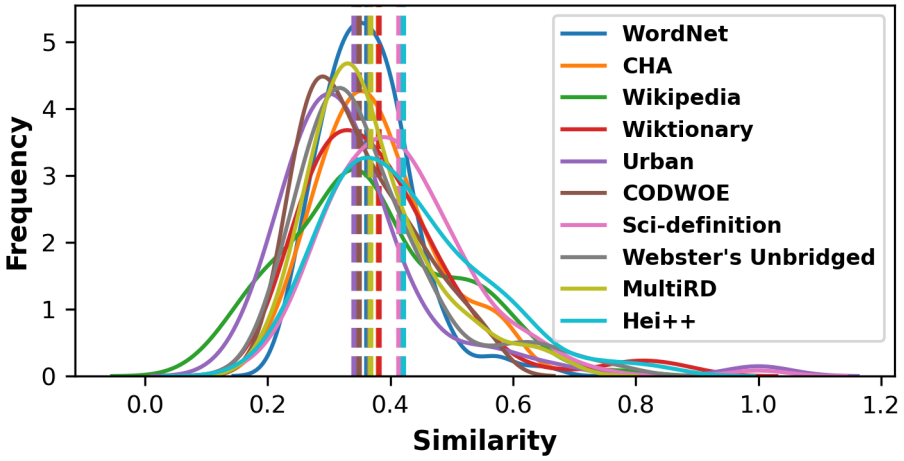
The second graph in the figure presents the similarity scores of the candidates gener-

ated by `gpt-4o-mini` for each source in the 3D-EX dataset. Some sources show clear patterns in candidate similarity. Wikipedia and Wiktionary have high variation, suggesting that their definitions are not always consistent, where some are clear while others are more ambiguous, leading the model to produce less focused predictions. Webster's Unabridged and Hei++ show more consistent and higher similarity, which suggests that their definitions help the model generate more related terms. In contrast, Urban Dictionary has the lowest similarity scores, likely because its informal and creative language makes it harder for the model to understand the intended meaning. Sci-definition stands out with the highest similarity, showing that technical definitions can guide the model to generate closely related, domain-specific terms.

For the final analysis, we examine the correlation between GEAR performance and the diversity (or similarity) of the terms generated by the LLMs (`gpt-4o-mini` and `Llama 3.1`), as shown in Figure 5.4. This is tested on Hill's dataset by examining the relationship between accuracy@1/10/100 and the average cosine similarity of candidates generated by both models. The cosine similarities were previously computed for candidates of 100 random definitions, and their averages are used in this analysis. Since higher similarity indicates lower diversity, the Pearson correlation coefficients for `gpt-4o-mini` (0.59) and `Llama 3.1` (0.57) reveal a moderate inverse relationship between accuracy and diversity. Recall that the aim of this analysis was to explore whether more diverse term generation could lead to a better input query embedding for the average and rank steps; however, this does not seem to be the case and, in fact, the more semantically compact (i.e., less variability in the generated terms) generated candidates clearly lead to better performance. These findings align with the similarity histogram (Figure 5.3a), which shows that the description prompt generates the least diverse candidates, reinforcing the connection between higher similarity and better performance. For future work, it would be interesting to explore if this holds also when longer lists of candidates are generated, or if the LLM is specifically instructed to generate diverse yet relevant terms given the definition.



(a) Cosine similarity scores between candidates in Hill's dataset (using Base Prompt 2)

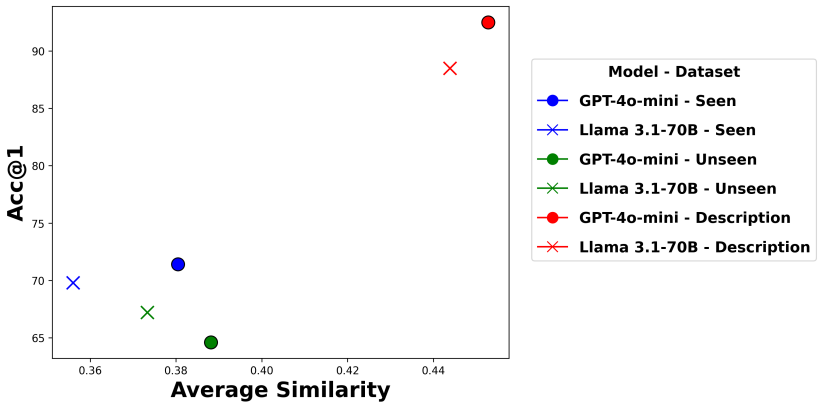


(b) Cosine similarity scores between candidates generated by gpt-4o-mini in 3D-EX dataset (using Reasoning Prompt)

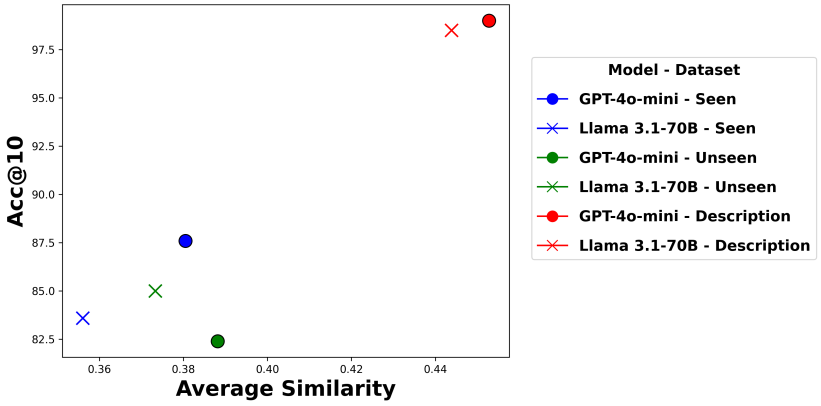
Figure 5.3: Cosine similarity between candidates generated by the LLMs. Dashed vertical lines indicate the mean similarity score for each case/dataset.

5.8 Summary and Limitations

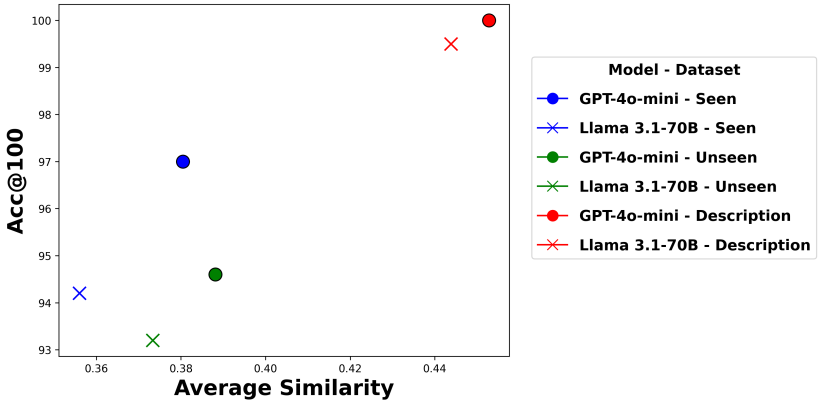
In this chapter, we introduced a simple yet effective method for reverse dictionary (RD) tasks. Our approach uses a large language model (LLM) to generate candidate terms



(a) Accuracy@1



(b) Accuracy@10



(c) Accuracy@100

Figure 5.4: Correlation between GEAR performance and candidates diversity in Hill's dataset.

given a definition, followed by embedding these candidates using pooling techniques, with simple averaging yielding the best results. This unsupervised method outperforms existing supervised methods on a well-known dataset, except for the *seen* split where some supervised methods appear to overfit. We also explored different components of this method and evaluated its performance across various dictionaries and evaluation settings.

However, several limitations remain. First, assuming that a single embedding model performs effectively across different registers and audiences may oversimplify the complex nature of language use. Considering different pre-training strategies could provide more nuanced insights into dictionary performance across contexts. Second, while the performance of GEAR was evaluated on the `3D-EX` dataset, a direct comparison with baseline RD models is necessary for a more complete evaluation. Third, the current approach averages all generated terms equally, which may limit prediction accuracy; task-specific weighted averaging could better capture the relative importance of terms. Finally, this work focuses primarily on English and does not address multilingual or cross-lingual generalization.

Chapter 6

Conclusions and Future Work

6.1 Introduction

This final chapter summarizes the research conducted in this thesis. It begins by linking the key contributions to the hypothesis and summarizing the major results in Section 6.2. Following this, we review the research questions addressed throughout the work in Section 6.3, and it ends by suggesting some ideas for future work in Section 6.4.

6.2 Thesis Summary and Contributions

The primary aim of this thesis was to conduct an in-depth investigation into the interplay between LRs and NLP tasks, exploring how LRs influence the effectiveness and outcomes of various NLP applications. Several previous studies have demonstrated the importance of LRs in improving NLP task performance; however, they have not thoroughly assessed the quality of these resources or compared their effectiveness with LLMs in handling more complex language tasks. This motivates us to bridge these gaps by providing a comprehensive evaluation and comparison.

The research hypothesis for this thesis was presented in Chapter 1. To remind the reader, the hypothesis is: “Dictionary examples serve as essential tools for elucidating the contextual and semantic meanings of terms. Within the scope of lexical resources,

these examples play an important role in providing clarity and depth to language understanding. The main hypothesis underpinning this thesis is that the quality of dictionary examples significantly influences the performance of NLP models that rely on word- and sentence-level semantics. Specifically, it suggests that when dictionary examples are detailed and informative, they empower NLP models to better understand term usage and meanings." In Chapter 2, we presented a general overview of LRs and their key components, with a particular focus on WN, the primary LR used in this thesis. Additionally, we reviewed existing literature on the interplay between LRs and NLP tasks.

In Chapter 3, we introduced the first experiment in this thesis that evaluates WN, especially its examples, through two main evaluation methods. The intrinsic evaluation is based on GDEX (Good Dictionary EXamples) criteria, and involves both automatic and human assessments, with CHA —a dataset based on the Oxford Dictionary— and ChatGPT-generated examples used for comparison. The extrinsic evaluation focuses on WN's application in the DM and word similarity tasks. We concluded that WN does not appear to be an optimal resource when informative dictionary examples are needed for downstream tasks, such as DM or deriving contextualized representations. These findings were later supported by Giulianelli et al. [81].

In Chapter 4, we introduced a new dataset, *3D-EX*, which integrates various English dictionaries and encyclopedias into one centralized knowledge repository. This dataset is derived from ten different sources, each with different styles, and went through a cleaning and unification process to form a standardized format of `<term, definition>` or `<term, definition, example>` triples. To assess the usefulness of this dataset, we conducted intrinsic experiments focusing on source classification and RD tasks. In the source classification task, the objective is to predict the original source of a given <term,definition> instance, helping to identify which sources are more distinctive and easier to classify. In the RD task, the goal is to retrieve a ranked list of the most relevant words based on a given definition.

In Chapter 5, we proposed *GEAR* which is a new, lightweight, and unsupervised method for RD. It begins with the Generate step, where LLMs generate a list of possible terms based on an input definition. In the Embed step, an embedding model converts these terms into vector embeddings. Next, in the Average step, the average of these embeddings is calculated to obtain a single vector for the generated terms. Finally, in the Rank step, this average embedding is compared with the embeddings of all terms in the dataset to get the top nearest neighbors. We evaluated GEAR on the three test sets from Hill et al. [93], the most widely used dataset for assessing RD models, to compare GEAR with existing RD methods. Our results show that GEAR outperforms all baselines on both the unseen definition set and the description set, only falling short on the seen split, likely due to overfitting in some methods. Additionally, we analyzed different components of GEAR and assessed its performance across various dictionaries and evaluation settings.

6.3 Research Questions

In this section the research questions will be revisited and discussed in terms of the relation between each question and the research that was conducted in this thesis.

Research Question 1: How does the quality and length of dictionary examples within LRs, such as WordNet, impact the performance of NLP models in tasks like definition modeling?

To address this question, we fine-tuned a BART-based model on WN and CHA datasets for definition generation, as detailed in Chapter 3, Section 3.4.1. We then evaluated the generated definitions using automatic string matching metrics: BLEU, METEOR, and ROUGE. Our findings indicate that the overall scores for the definition generation model based on WN examples are generally low, particularly when compared to the model trained on CHA examples. Furthermore, an analysis of the definitions gener-

ated by the WN-trained model shows notable problems such as under-specificity and circularity.

Research Question 2: How can embeddings for words, phrases, and sentences be improved by leveraging dictionary examples?

To answer this question, we conducted a similarity experiment described in Chapter 3, Section 2.3.3.1, where we evaluated examples from WN against CHA and those generated by ChatGPT. This evaluation involved measuring the similarity between word pairs using these three sets of examples, as informative examples are expected to produce higher-quality contextual embeddings. Our findings indicate that WN examples, which tend to be less informative, too short, and often contain highly ambiguous words, result in word embeddings that perform significantly worse on word similarity benchmarks compared to those generated from CHA and ChatGPT examples.

Research Question 3: How does the integration of multiple lexical resources into a centralized knowledge repository contribute to improving NLP models across different downstream tasks?

As demonstrated in Chapter 4, creating a centralized repository enhances NLP models by enabling them to interpret language across various contexts. By combining diverse sources such as dictionaries and linguistic databases, the repository provides a richer and more detailed understanding of language. This variety helps models better capture linguistic patterns and meanings, resulting in improved performance.

Research Question 4: Can the task of reverse dictionary improve by combining "the best of both worlds", i.e., LLM generation capabilities as well as semantic similarities derived from dictionary embeddings?

To address this, we employed a simple unsupervised RD method that leverages LLMs (see Chapter 5) to generate candidate terms from given definitions and embed them

using a pooling technique. This approach outperforms existing supervised methods, highlighting the effectiveness of this combined method.

6.4 Future Work

In this section, we discuss some of the possible ways in which the research in this thesis could be extended further in the future.

Evaluating WN extrinsically in different NLP tasks We would like to explore extrinsic evaluations of WN across various NLP tasks to assess its effectiveness in different applications. In addition, we could conduct experiments comparing WN's performance with other datasets—beyond the Oxford Dictionary and ChatGPT—and language models in different areas of NLP.

WN human evaluation We aim to extend the questionnaire to include other resources and LLMs, leveraging the scores we obtained for training dictionary scoring systems, which we believe would be valuable tools for both lexicographers and NLP practitioners. Additionally, considering the intended users of the dictionary is important when comparing dictionaries, as this can influence the complexity of the examples.

Exploring 3D-EX We aim to further explore the potential of 3D-EX for downstream NLP tasks by incorporating more resources and exploring multilingual variants. An additional avenue would be to examine the interaction of unorthodox dictionaries, such as Urban, with traditional lexicographic resources in the context of controlled technical or jargon-driven domain modeling. Furthermore, future work could focus on developing an automated workflow to identify and integrate new resources into 3D-EX using AI agents, ensuring the dataset remains up to date and comprehensive.

Enhancing GEAR with Different LLMs and Multilingual Capabilities We intend to explore various LLMs for the generation step, potentially developing a task-specific weighted average approach that could be learned using simple neural network architectures. This is similar to the method used by Wang et al. [209], who employed word classification datasets to tune contextualized word embeddings. Additionally, expanding GEAR's functionality to support multiple languages would be beneficial, leveraging multilingual encoders such as XLM-R or mBERT.

Evaluating LLMs generated examples through GEAR We plan to evaluate the examples generated from different LLMs by applying the *reasoning prompt* in GEAR method based on GDEX criteria, which include both automatic and human evaluation, as was implemented in in Chapter 3 that evaluates WN's examples. Additionally, we will assess the similarity of the generated examples to the target definitions, focusing on their relevance to the intended meaning.

Bibliography

[1] A. B. Abacha and D. Demner-Fushman. A question-entailment approach to question answering. *BMC Bioinformatics*, 20(1), oct 2019. doi: 10.1186/s12859-019-3119-4.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] A. Adamska-Sałaciak. Dictionary definitions: Problems and solutions. *Studia Linguistica Universitatis Iagellonicae Cracoviensis*, 129(4), 2015.

[4] E. Agirre and P. Edmonds. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media, 2007.

[5] D. Alfarone and J. Davis. Unsupervised learning of an is-a taxonomy from a limited domain-specific corpus. In *International Joint Conference on Artificial Intelligence*, 2015. URL `https://api.semanticscholar.org/CorpusID:16493076`.

[6] F. Almeman and L. Espinosa Anke. Putting WordNet's dictionary examples in the context of definition modelling: An empirical analysis. In M. Zock, E. Chersoni, Y.-Y. Hsu, and E. Santus, editors, *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 42–48, Taipei, Taiwan, Nov. 2022. As-

sociation for Computational Linguistics. URL https://aclanthology.org/2022.cogalex-1.6.

[7] F. Almeman and L. Espinosa-Anke. Gear: A simple generate, embed, average and rank approach for unsupervised reverse dictionary, 2024. URL https://arxiv.org/abs/2412.06654.

[8] F. Almeman, H. Sheikhi, and L. Espinosa Anke. 3D-EX: A unified dataset of definitions and dictionary examples. In R. Mitkov and G. Angelova, editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 69–79, Varna, Bulgaria, Sept. 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.ranlp-1.8.

[9] F. Almeman, S. Schockaert, and L. Espinosa Anke. WordNet under scrutiny: Dictionary examples in the era of large language models. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17683–17695, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1538.

[10] R. A. Amsler. *The structure of the Merriam-Webster pocket dictionary*. The University of Texas at Austin, 1980.

[11] M. Apidianaki and A. G. Soler. All dolphins are intelligent and some are friendly: Probing bert for nouns' semantic properties and their prototypicality. *arXiv preprint arXiv:2110.06376*, 2021.

[12] B. S. Atkins and M. Rundell. *The Oxford guide to practical lexicography*. Oxford University Press, 2008.

[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web: 6th International*

Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings, pages 722–735. Springer, 2007.

[14] T. August, C. Reinecke, and N. Smith. generating scientific definitions with controllable commplexity. 2022.

[15] D. Bahdanau, T. Bosc, S. Jastrzębski, E. Grefenstette, P. Vincent, and Y. Bengio. Learning to compute word embeddings on the fly, 2018.

[16] L. Bajčetić and T. Declerck. Using Wiktionary to create specialized lexical resources and datasets. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3457–3460, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.370.

[17] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada, Aug. 1998. Association for Computational Linguistics. doi: 10.3115/980845.980860. URL https://aclanthology.org/P98-1013/.

[18] M. Bakrey. All about lexicons in nlp. https://mohamedbakrey094.medium.com/all-about-lexicons-in-nlp-12ada00c2821, 2023. Accessed: 2024-09-21.

[19] E. Barba, L. Procopio, C. Lacerra, T. Pasini, and R. Navigli. Exemplification modeling: Can you give me an example, please? In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3779–3785. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/520. URL https://doi.org/10.24963/ijcai.2021/520. Main Track.

[20] H. Bejoint. The bloomsbury companion to lexicography edited by howard jackson. *Dictionaries: Journal of the Dictionary Society of North America*, 35: 374–381, 01 2014. doi: 10.1353/dic.2014.0017.

[21] M. Bevilacqua, M. Maru, and R. Navigli. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.585. URL https://aclanthology.org/2020.emnlp-main.585.

[22] S. Bila, W. Watanabe, T. Hashimoto, T. Tokunaga, and H. Tanaka. Dictionary search based on the target word description. 01 2004.

[23] T. Blevins, M. Joshi, and L. Zettlemoyer. Fews: Large-scale, low-shot word sense disambiguation with the dictionary, 2021.

[24] G. Boella and L. Di Caro. Extracting definitions and hypernym relations relying on syntactic dependencies and support vector machines. In H. Schuetze, P. Fung, and M. Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 532–537, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-2095.

[25] F. Bond and R. Foster. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1133.

[26] G. Bordea, E. Lefever, and P. Buitelaar. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In S. Bethard, M. Carpuat, D. Cer, D. Jurgens, P. Nakov, and T. Zesch, editors, *Proceedings of the 10th International*

Workshop on Semantic Evaluation (SemEval-2016), pages 1081–1091, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1168. URL https://aclanthology.org/S16-1168.

[27] T. Bosc and P. Vincent. Auto-encoding dictionary definitions into consistent word embeddings. In *EMNLP*, pages 1522–1532, 2018.

[28] T. Brants and A. Franz. Web 1t 5-gram version 1 (2006). *Linguistic Data Consortium, Philadelphia*, 2006.

[29] B. Broda, M. Piasecki, and S. Szpakowicz. A wordnet from the ground up. *Oficyna Wydawnicza Politechniki Wrocławskiej*, 01 2009.

[30] E. Bruni, N. K. Tran, and M. Baroni. Multimodal distributional semantics. *J. Artif. Intell. Res.*, 49:1–47, 2014. URL https://api.semanticscholar.org/CorpusID:2618475.

[31] A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*, volume 2, pages 2–2, 2001.

[32] J. A. Burgess. When is circularity in definitions benign? *The Philosophical Quarterly*, 58(231):214–233, 2008.

[33] B. Cai, N. Clarence, D. Liang, and S. Hotama. Low-cost generation and evaluation of dictionary example sentences. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3538–3549, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.194. URL https://aclanthology.org/2024.naacl-long.194.

[34] J. Camacho-Collados, L. Espinosa Anke, and M. T. Pilehvar. The interplay between lexical resources and natural language processing. In *Proceed-*

ings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts, pages 17–23, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-6004. URL https://aclanthology.org/N18-6004.

[35] M. Ceccato, N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry. Ambiguity identification and measurement in natural language texts. 2004.

[36] Y. Chae and T. Davidson. Large language models for text classification: From zero-shot learning to fine-tuning. *Open Science Foundation*, 2023.

[37] T.-Y. Chang and Y.-N. Chen. What does this word mean? explaining contextualized embeddings with natural language definition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6064–6070, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1627. URL https://aclanthology.org/D19-1627.

[38] T.-Y. Chang, T.-C. Chi, S.-C. Tsai, and Y.-N. Chen. xsense: Learning sense-separated sparse representations and textual definitions for explainable word sense networks, 2018.

[39] G. Chen and J. Su. Towards non-ambiguous reverse dictionary. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1113–1120. IEEE, 2021.

[40] H. Chen. String metrics and word similarity applied to information retrieval. Master's thesis, Itä-Suomen yliopisto, 2012.

[41] J. Chen and J. Liu. Combining ConceptNet and WordNet for word sense disambiguation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 686–694, Chiang Mai, Thailand, Nov. 2011. Asian

Federation of Natural Language Processing. URL https://aclanthology.org/I11-1077.

[42] P. Chen and Z. Zhao. A unified model for reverse dictionary and definition modelling. *arXiv preprint arXiv:2205.04602*, 2022.

[43] Q. Chen, F.-L. Li, G. Xu, M. Yan, J. Zhang, and Y. Zhang. Dictbert: Dictionary description knowledge enhanced language model pre-training via contrastive learning. *arXiv preprint arXiv:2208.00635*, 2022.

[44] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

[45] G. Chowdhury. Natural language processing. *ARIST*, 37:51–89, 01 2005. doi: 10.1002/aris.1440370103.

[46] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[47] M. Coleman and T. L. Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283, 1975.

[48] W. contributors. Wikipedia: The free encyclopedia, 2001. URL `https://www.wikipedia.org`.

[49] A. Cowie. Language as words: Lexicography. 1990.

[50] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, system demonstrations*, pages 97–102, 2017.

[51] E. Dale and J. S. Chall. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54, 1948.

[52] C. Delli Bovi, L. Telesca, and R. Navigli. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543, 2015.

[53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[54] U. Dhungana, S. Shakya, K. Baral, and B. Sharma. Word sense disambiguation using wsd specific wordnet of polysemy words. *International Journal on Natural Language Computing*, 3, 09 2014. doi: 10.5121/ijnlc.2014.3405.

[55] D. A. Dillman, J. D. Smyth, and L. M. Christian. *Internet, Phone, Mail, and Mixed-Mode Surveys: The Tailored Design Method*. Wiley, 4 edition, 2014.

[56] Education Northwest. What are the traits?, 2020.

[57] L. Espinosa-Anke, J. Camacho-Collados, S. Rodríguez-Fernández, H. Saggion, and L. Wanner. Extending wordnet with fine-grained collocational information via supervised distributional learning. 12 2016.

[58] L. Espinosa-Anke, H. Saggion, F. Ronzano, and R. Navigli. Extasem! extending, taxonomizing and semantifying domain terminologies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[59] L. Espinosa Anke, J. Codina-Filba, and L. Wanner. Evaluating language models for the retrieval and categorization of lexical collocations. In P. Merlo, J. Tiedemann, and R. Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1406–1417, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.120. URL https://aclanthology.org/2021.eacl-main.120.

[60] L. Espinosa-Anke, A. Shvets, A. Mohammadshahi, J. Henderson, and L. Wanner. Multilingual extraction and categorization of lexical collocations with graph-aware transformers. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 89–100, 2022.

[61] L. Espinosa-Anke, A. Shvets, A. Mohammadshahi, J. Henderson, and L. Wanner. Multilingual extraction and categorization of lexical collocations with graph-aware transformers, 2022. URL https://arxiv.org/abs/2205.11456.

[62] K. Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings, 2019.

[63] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346.

[64] M. Farouk. Measuring text similarity based on structure and word embedding. *Cognitive Systems Research*, 63:1–10, 2020. ISSN 1389-0417. doi: https://doi.org/10.1016/j.cogsys.2020.04.002. URL https://www.sciencedirect.com/science/article/pii/S1389041720300218.

[65] M. Faruqui, J. Dodge, S. Jauhar, C. Dyer, E. Hovy, and N. Smith. Retrofitting word vectors to semantic lexicons. 11 2014. doi: 10.3115/v1/N15-1184.

[66] A. R. Feinstein and D. V. Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990. doi: 10.1016/0895-4356(90)90158-L.

[67] C. Fellbaum and C. F. Baker. Can WordNet and FrameNet be made ''interoperable''? In *Proceedings of The First International Conference on Global Interoperability for Language Resources*, pages 67–74, 2008. URL `http://icgl.ctl.cityu.edu.hk/2008/html/resources/~proceeding_conference.pdf`.

[68] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic BERT sentence embedding. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.62. URL `https://aclanthology.org/2022.acl-long.62`.

[69] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. doi: 10.1037/h0031619.

[70] A. Frankenberg-Garcia. Learners' use of corpus examples. *International Journal of Lexicography*, 25(3):273–296, 2012.

[71] A. Frankenberg-Garcia. The use of corpus examples for language comprehension and production. *ReCALL*, 26(2):128–146, 2014. doi: 10.1017/S0958344014000093.

[72] A. Frankenberg-Garcia, R. Lew, J. C. Roberts, G. P. Rees, and N. Sharma. Developing a writing assistant to help eap writers with collocations in real time. *ReCALL*, 31(1):23–39, 2019.

[73] A. A. Freihat, H. M. Khalilia, G. Bella, and F. Giunchiglia. Advancing the Arabic WordNet: Elevating content quality. In H. Al-Khalifa, K. Darwish, H. Mubarak, M. Ali, and T. Elsayed, editors, *Proceedings of the 6th Workshop*

on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation @ LREC-COLING 2024, pages 74–83, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.osact-1.9/.

[74] A. Gadetsky, I. Yakubovskiy, and D. Vetrov. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2043. URL https://aclanthology.org/P18-2043.

[75] A. Gajbhiye, L. Espinosa-Anke, and S. Schockaert. Modelling commonsense properties using pre-trained bi-encoders, 2022. URL https://arxiv.org/abs/2210.02771.

[76] A. Gajbhiye, Z. Bouraoui, L. Espinosa Anke, and S. Schockaert. AMenDeD: Modelling concepts by aligning mentions, definitions and decontextualised embeddings. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 801–811, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.72.

[77] L. Galke, A. Saleh, and A. Scherp. Word embeddings for practical information retrieval. In *Informatik 2017*, pages 2155–2167. Gesellschaft für Informatik, 2017.

[78] J. Ganitkevitch, B. Van Durme, and C. Callison-Burch. PPDB: The paraphrase database. In L. Vanderwende, H. Daumé III, and K. Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764,

Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1092/.

[79] N. Gardner, H. Khan, and C.-C. Hung. Definition modeling: literature review and dataset analysis. *Applied Computing and Intelligence*, 2(1):83–98, 2022. ISSN 2771-392X. doi: 10.3934/aci.2022005. URL https://www.aimspress.com/article/doi/10.3934/aci.2022005.

[80] D. Gerz, I. Vulić, F. Hill, R. Reichart, and A. Korhonen. SimVerb-3500: A large-scale evaluation set of verb similarity, Nov. 2016. URL https://aclanthology.org/D16-1235.

[81] M. Giulianelli, I. Luden, R. Fernandez, and A. Kutuzov. Interpretable word sense representations via definition generation: The case of semantic change analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.176. URL https://aclanthology.org/2023.acl-long.176.

[82] T. T. Goh, N. A. A. Jamaludin, H. Mohamed, M. N. Ismail, and H. Chua. Semantic similarity analysis for examination questions classification using wordnet. *Applied Sciences*, 13(14), 2023. ISSN 2076-3417. doi: 10.3390/app13148323. URL https://www.mdpi.com/2076-3417/13/14/8323.

[83] A. Gómez-Pérez. Ontological engineering: A state of the art. *Expert update*, 6 (3):33–43, 2003.

[84] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[85] M. Günther, L. Milliken, J. Geuter, G. Mastrapas, B. Wang, and H. Xiao. Jina embeddings: A novel set of high-performance sentence embedding models, 2023.

[86] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 2023.

[87] M. Hartung. *Distributional Semantic Models of Attribute Meaning in Adjectives and Nouns*. PhD thesis, 12 2015.

[88] C. Havasi, R. Speer, and J. Alonso. Conceptnet 3: A flexible, multilingual semantic network for common sense knowledge. *Proceedings of Recent Advances in Natural Language Processing*, 01 2007.

[89] X. He and S. M. Yiu. Controllable dictionary example generation: Generating example sentences for specific targeted audiences. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 610–627, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.46. URL https://aclanthology.org/2022.acl-long.46.

[90] A. Healey. The oxford handbook of lexicography ed. by philip durkin. *Dictionaries: Journal of the Dictionary Society of North America*, 38:114–119, 01 2017. doi: 10.1353/dic.2017.0013.

[91] V. Henrich, E. Hinrichs, and T. Vodolazova. Webcage: a web-harvested corpus annotated with germanet senses. pages 387–396, 04 2012.

[92] F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation, 2015.

[93] F. Hill, K. Cho, A. Korhonen, and Y. Bengio. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016.

[94] E. Hovy, R. Navigli, and S. P. Ponzetto. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194: 2–27, 2013.

[95] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

[96] H. Huang, T. Kajiwara, and Y. Arase. Definition modelling for appropriate specificity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2499–2509, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.194. URL https://aclanthology.org/2021.emnlp-main.194.

[97] J. Huang, H. Shao, and K. C.-C. Chang. Cdm: Combining extraction and generation for definition modeling, 2021.

[98] F. Ilievski, P. Szekely, J. Cheng, F. Zhang, and E. Qasemi. Consolidating commonsense knowledge, 2020.

[99] S. Ishiwatari, H. Hayashi, N. Yoshinaga, G. Neubig, S. Sato, M. Toyoda, and M. Kitsuregawa. Learning to describe unknown phrases with local and global contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3467–3476, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1350. URL https://aclanthology.org/N19-1350.

[100] M. Jakubíček, M. Rundell, C. Breathnach, and P. Mianáin. Dante resurrected large open source lexical data database for english, 10 2024.

[101] S. Johnson. *A Dictionary of the English Language*. URL `https://johnsonsdictionaryonline.com`. 1755, 1773. Edited by Beth Rapp Young, Jack Lynch, William Dorner, Amy Larner Giroux, Carmen Faye Mathes, and Abigail Moreshead. 2021.

[102] M. Joshi, K. Lee, Y. Luan, and K. Toutanova. Contextualized representations using textual encyclopedic knowledge. *arXiv preprint arXiv:2004.12006*, 2020.

[103] K. Kann, S. Rothe, and K. Filippova. Sentence-level fluency evaluation: References help, but can be spared! In A. Korhonen and I. Titov, editors, *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1031. URL `https://aclanthology.org/K18-1031/`.

[104] G. Kanumolu, L. Madasu, P. Baswani, A. Mukherjee, and M. Shrivastava. Unsupervised approach to evaluate sentence-level fluency: Do we really need reference?, 2023. URL `https://arxiv.org/abs/2312.01500`.

[105] D. Kartsaklis, M. T. Pilehvar, and N. Collier. Mapping text to knowledge graph entities using multi-sense LSTMs. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1959–1970, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1221. URL `https://aclanthology.org/D18-1221`.

[106] A. Kilgarriff. A detailed, accurate, extensive, available english lexical database. pages 21–24, 06 2010.

[107] A. Kilgarriff, P. Rychlý, P. Smrz, and D. Tugwell. The sketch engine. In G. Williams and S. Vessier, editors, *Proceedings of the 11th EURALEX International Congress*, pages 105–115, Lorient, France, july 2004. ISBN 29-52245-70-3.

[108] A. Kilgarriff, M. Husák, K. McAdam, M. Rundell, and P. Rychlý. Gdex: Automatically finding good dictionary examples in a corpus. In J. D. Elisenda Bernal, editor, *Proceedings of the 13th EURALEX International Congress*, pages 425–432, Barcelona, Spain, jul 2008. Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra. ISBN 978-84-96742-67-3.

[109] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch, 1975.

[110] C. Kong, Y. Wang, R. Chong, L. Yang, H. Zhang, E. Yang, and Y. Huang. Blcuicall at semeval-2022 task 1: Cross-attention multitasking framework for definition modeling. *arXiv preprint arXiv:2204.07701*, 2022.

[111] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

[112] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. doi: 10.2307/2529310.

[113] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. pages 228–231, 07 2007.

[114] M. C. Lee, J. W. Chang, and T. C. Hsieh. A grammar-based semantic similarity algorithm for natural language sentences. *The Scientific World Journal*, 2014 (1):437162, 2014.

[115] O. Levy, S. Remus, C. Biemann, and I. Dagan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 970–976, 2015.

[116] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 2019. doi: 10.18653/v1/2020.acl-main.703.

[117] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li. On the sentence embeddings from pre-trained language models, 2020. URL https://arxiv.org/abs/2011.05864.

[118] X. Li and J. Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

[119] Y. Li, D. McLean, Z. A. Bandar, J. D. O'shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150, 2006.

[120] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[121] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

[122] T. Lino and L. Coheur. Challenges on the automatic translation of collocations. *MULTI-WORD UNITS IN MACHINE TRANSLATION AND TRANSLATION TECHNOLOGIES MUMTTT2015*, page 74, 2015.

[123] F. Liu, I. Vulić, A. Korhonen, and N. Collier. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders, 2021. URL https://arxiv.org/abs/2104.08027.

[124] Q. Liu, F. Liu, N. Collier, A. Korhonen, and I. Vulić. Mirrorwic: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25rd Conference on Computational Natural Language Learning (CoNLL)*, 2021.

[125] S. Liu, C. Gao, and Y. Li. Large language model agent for hyper-parameter optimization, 2024. URL https://arxiv.org/abs/2402.01881.

[126] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[127] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. S. Weld. S2orc: The semantic scholar open research corpus, 2020.

[128] S. B. Mane, H. N. Patil, K. B. Madaswar, and P. N. Sadavarte. Wordalchemy: a transformer-based reverse dictionary. In *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pages 1–5. IEEE, 2022.

[129] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18 (1):50–60, 1947.

[130] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís. Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards Interfaces*, 35(5):482–489, 2013. ISSN 0920-5489. doi: https://doi.org/10.1016/j.csi.2012.09.004. URL https://www.sciencedirect.com/science/article/pii/S0920548912001080.

[131] J. Mccrae, A. Rademaker, E. Rudnicka, and F. Bond. English wordnet 2020: Improving and extending a wordnet for english using an open-source methodology. 09 2020.

[132] J. P. McCrae, C. Chiarcos, P. Cimiano, J. Gracia, S. Hellmann, Y. Isaakidis, C. Kiefer, and et al. Linguistic linked data: Representation, generation and applications. In P. Hitzler, K. Janowicz, and V. Sabol, editors, *The Semantic Web: Semantics and Big Data*, pages 157–170. Springer, 2019.

[133] S. Melacci, A. Globo, and L. Rigutini. Enhancing modern supervised word sense disambiguation models by semantic lexical resources, 2024. URL https://arxiv.org/abs/2402.13302.

[134] R. Merx, E. Vylomova, and K. Kurniawan. Generating bilingual example sentences with large language models as lexicography assistants, 2024. URL https://arxiv.org/abs/2410.03182.

[135] T. Mickus, D. Paperno, and M. Constant. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland, Sept. 2019. Linköping University Electronic Press. URL https://aclanthology.org/W19-6201.

[136] T. Mickus, D. Paperno, and M. Constant. Mark my word: A sequence-to-sequence approach to definition modeling, 2019.

[137] T. Mickus, K. Van Deemter, M. Constant, and D. Paperno. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.semeval-1.1. URL https://aclanthology.org/2022.semeval-1.1.

[138] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013. URL https://api.semanticscholar.org/CorpusID:5959482.

[139] G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–, 11 1995. doi: 10.1145/219717.219748.

[140] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey, 2024. URL https://arxiv.org/abs/2402.06196.

[141] S. R. Mirhoseini, F. Vahedi, and J. Nasiri. E-mail phishing detection using natural language processing and machine learning techniques. 06 2020.

[142] J. Morato, M. Marzal, J. Llorens, and J. Moreiro. Wordnet applications. *Proceedings of the 2nd Global Wordnet Conference*, 2004, 04 2004.

[143] Y. Morinaga and K. Yamaguchi. *Improvement of Reverse Dictionary by Tuning Word Vectors and Category Inference: 24th International Conference, ICIST 2018, Vilnius, Lithuania, October 4–6, 2018, Proceedings*, pages 533–545. 08 2018. ISBN 978-3-319-99971-5. doi: 10.1007/978-3-319-99972-2_44.

[144] G. R. Mtallo. A critical study of the word meanings in dictionaries: a case of oxford advanced learners dictionary. 2015.

[145] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, 2023.

[146] R. Müller, S. Kornblith, and G. Hinton. When does label smoothing help?, 2020. URL https://arxiv.org/abs/1906.02629.

[147] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian. A comprehensive overview of large language models, 2024. URL https://arxiv.org/abs/2307.06435.

[148] R. Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[149] R. Navigli and S. Ponzetto. Babelnet: Building a very large multilingual semantic network. pages 216–225, 09 2010.

[150] R. Navigli and S. P. Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence*, 193:217–250, 2012.

[151] R. Navigli and P. Velardi. Learning word-class lattices for definition and hypernym extraction. In J. Hajič, S. Carberry, S. Clark, and J. Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1134.

[152] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space, 2015.

[153] T.-P. Nguyen, S. Razniewski, J. Romero, and G. Weikum. Refined commonsense knowledge from large-scale web contents. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–16, 2022. doi: 10.1109/tkde.2022.3206505.

[154] K. Ni and W. Y. Wang. Learning to explain non-standard English words and phrases. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 413–417, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-2070.

[155] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the international conference on formal ontology in information systems-Volume 2001*, pages 2–9. ACM, 2001.

[156] T. Noraset, C. Liang, L. Birnbaum, and D. Downey. Definition modeling: Learning to define word embeddings in natural language. pages 3259–3266, 2017.

[157] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

[158] Y. Park, R. J. Byrd, and B. Boguraev. Automatic glossary extraction: Beyond terminology identification. In *International Conference on Computational Linguistics*, 2002. URL https://api.semanticscholar.org/CorpusID:267872215.

[159] A. Peckham. Urban dictionary, 1999. URL https://www.urbandictionary.com.

[160] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

[161] G. Pereyra, G. Tucker, J. Chorowski, Łukasz Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions, 2017. URL https://arxiv.org/abs/1701.06548.

[162] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations, 2018.

[163] M. E. Peters, M. Neumann, R. L. L. IV, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith. Knowledge enhanced contextual word representations, 2019. URL https://arxiv.org/abs/1909.04164.

[164] T. Phan. Applying wordnet in teaching the lexical semantics of english nouns. *Science Technology Development Journal*, 27, 11 2024. doi: 10.32508/stdj.v27iSI.4411.

[165] S. Piao, P. Rayson, D. Archer, and A. McEnery. Evaluating lexical resources for a semantic tagger. In *Proceedings of LREC 2004*, pages 499–502. European Language Resources Association, 2004.

[166] M. T. Pilehvar. On the importance of distinguishing word meaning representations: A case study on reverse dictionary mapping. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2151–2156, 2019.

[167] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. 2019. URL https://openai.com/blog/better-language-models/.

[168] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[169] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

[170] K. Raju. Speech based voice recognition system for natural language processing. *International Journal of Computer Science and Information Technogy*, 08 2014.

[171] A. Ranjan Pal and D. Saha. Word sense disambiguation: A survey. *International Journal of Control Theory and Computer Modeling*, 5(3):1–16, Jul 2015. ISSN 2249-1155. doi: 10.5121/ijctcm.2015.5301. URL http://dx.doi.org/10.5121/ijctcm.2015.5301.

[172] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020. URL https://api.semanticscholar.org/CorpusID:221191193.

[173] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*, pages 177–185. Springer, 2016.

[174] G. A. Recski, E. Iklódi, K. A. Pajkossy, and A. Kornai. Measuring semantic similarity of words using concept networks. Association for Computational Linguistics, 2016.

[175] M. Reid, E. Marrese-Taylor, and Y. Matsuo. Vcdm: Leveraging variational bi-encoding and deep contextualized word representations for improved definition modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6331–6344, 2020.

[176] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[177] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conference on Artificial Intelligence*, 1995. URL https://api.semanticscholar.org/CorpusID:1752785.

[178] S. Rodríguez-Fernández, L. E. Anke, R. Carlini, and L. Wanner. Semantics-driven recognition of collocations using word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 499–505, 2016.

[179] E. S. Ruzzetti, L. Ranaldi, M. Mastromattei, F. Fallucchi, N. Scarpato, and F. M. Zanzotto. Lacking the embedding of a word? look it up into a traditional dictionary. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2651–2662, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.208. URL https://aclanthology.org/2022.findings-acl.208.

[180] M. Sap, R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi. Atomic: an atlas of machine commonsense for if-then reasoning. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33013027. URL https://doi.org/10.1609/aaai.v33i01.33013027.

[181] S. K. Sarma, D. Sarmah, B. Brahma, H. Bharali, M. Mahanta, and U. Saikia. Building multilingual lexical resources using wordnets: Structure, design and implementation. In *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon*, pages 161–170, 2012.

[182] J. Sauro and J. R. Lewis. *Quantifying the User Experience: Practical Statistics for User Research*. Morgan Kaufmann, 2 edition, 2016.

[183] V. Segonne, M. Candito, and B. Crabbé. Using Wiktionary as a resource for WSD : the case of French verbs. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 259–270, Gothenburg, Sweden, May 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-0422. URL https://aclanthology.org/W19-0422.

[184] L. K. Senel, T. Schick, and H. Schütze. Coda21: Evaluating language understanding capabilities of nlp models with context-definition alignment, 2022.

[185] S. C. Sereno, J. M. Pacht, and K. Rayner. The effect of meaning frequency on processing lexically ambiguous words: Evidence from eye fixations. *Psychological Science*, 3(5):296–301, 1992.

[186] V. Seretan and E. Wehrli. Collocation translation based on sentence alignment and parsing. In *Actes de la 14ème conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, pages 375–384, 2007.

[187] R. Shaw, A. Datta, D. VanderMeer, and K. Dutta. Building a scalable database-driven reverse dictionary. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):528–540, 2011.

[188] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27:443–460, 2015. URL `https://api.semanticscholar.org/CorpusID:16320392`.

[189] V. Shwartz, Y. Goldberg, and I. Dagan. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076*, 2016.

[190] R. Sinhal and K. Gupta. Machine translation approaches and design aspects. *IOSR Journal of Computer Engineering*, 16:22–25, 01 2014. doi: 10.9790/0661-16122225.

[191] M. D. E. Souci, Y. Cherifi, L. Berkani, M. S. H. Ameur, and A. Guessoum. Enrichment of Arabic WordNet using machine translation and transformers. In M. Abbas and A. A. Freihat, editors, *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 333–340, Online, Dec. 2023. Association for Computational Linguistics. URL `https://aclanthology.org/2023.icnlsp-1.36/`.

[192] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159. URL https://doi.org/10.2307/1412159.

[193] H. Su, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W.-t. Yih, N. A. Smith, L. Zettlemoyer, T. Yu, et al. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

[194] K.-T. Sun, Y.-M. Huang, and M.-C. Liu. A wordnet-based near-synonyms and similar-looking word learning system. *Educational Technology Society*, 14: 121–134, 01 2011.

[195] M. Taulé, M. García, N. Artigas, and M. Martí. Evaluating lexical resources for wsd. In *Euralex Proceedings*, 2004.

[196] R. Temmerman. *Towards new ways of terminology description: The sociocognitive approach*. John Benjamins Publishing, 2000.

[197] S. Tian, S. Huang, R. Li, and C. Wei. A prompt construction method for the reverse dictionary task of large-scale language models. *Engineering Applications of Artificial Intelligence*, 133:108596, 2024.

[198] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. URL https://api.semanticscholar.org/CorpusID:257219404.

[199] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[200] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394, 2010.

[201] Y. Varelas, E. Voutsakis, P. Raftopoulou, E. Petrakis, and E. Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. pages 10–16, 11 2005. doi: 10.1145/1097047.1097051.

[202] Various. *Webster's Unabridged Dictionary*. Project Gutenberg, 2009.

[203] P. Velardi, S. Faralli, and R. Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39:665–707, 09 2013. doi: 10.1162/COLI_a_00146.

[204] E. M. Voorhees. Natural language processing and information retrieval. In *International summer school on information extraction*, pages 32–48. Springer, 1999.

[205] I. Vulić, D. Gerz, D. Kiela, F. Hill, and A. Korhonen. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835, 2017.

[206] S. Walter and M. Pinkal. Automatic extraction of definitions from German court decisions. In M. E. Califf, M. A. Greenwood, M. Stevenson, and R. Yangarber, editors, *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 20–28, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `https://aclanthology.org/W06-0203`.

[207] Wang. Should they look it up? the role of dictionaries in language learning. 2001.

[208] T. Wang and G. Hirst. Exploring patterns in dictionary definitions for synonym extraction. *Natural Language Engineering*, 18(3):313–342, 2012.

[209] Y. Wang, Z. Bouraoui, L. E. Anke, and S. Schockaert. Deriving word vectors from contextualized language models using topic-aware mention selection. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 185–194, 2021.

[210] Y. Wang, Z. Bouraoui, L. Espinosa Anke, and S. Schockaert. Sentence selection strategies for distilling word embeddings from BERT. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2591–2600, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.277.

[211] N. Webster. *Webster's unabridged dictionary of the English language*. Kikwansha, 1900.

[212] L. Wenjun, W. Hailan, W. Xiping, and H. Mengshu. A word sense disambiguation method based on multiple sense graph. In Z. Nianyin and R. B. Pachori, editors, *Proceedings of 2024 International Conference on Machine Learning and Intelligent Computing*, volume 245 of *Proceedings of Machine Learning Research*, pages 40–47. PMLR, 26–28 Apr 2024. URL https://proceedings.mlr.press/v245/wenjun24a.html.

[213] P. West, X. Lu, N. Dziri, F. Brahman, L. Li, J. D. Hwang, L. Jiang, J. Fisher, A. Ravichander, K. Chandu, B. Newman, P. W. Koh, A. Ettinger, and Y. Choi. The generative ai paradox: "what it can create, it may not understand", 2023. URL https://arxiv.org/abs/2311.00059.

[214] E. Westerhout. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 61–67, 2009.

[215] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural

language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[216] J. Wu, Y. Chang, T. Mitamura, and J. Chang. Automatic collocation suggestion in academic writing. In *Proceedings of the ACL Conference, Short paper track*, Uppsala, 2010.

[217] H. Xu, Y. Chen, Z. Liu, Y. Wen, and X. Yuan. Taxoprompt: A prompt-based generation method with taxonomic context for self-supervised taxonomy expansion. In L. D. Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4432–4438. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/615. URL https://doi.org/10.24963/ijcai.2022/615. Main Track.

[218] N. Xu, Q. Zhang, M. Zhang, P. Qian, and X. Huang. On the tip of the tongue: Analyzing conceptual representation in large language models with reverse-dictionary probe. *arXiv preprint arXiv:2402.14404*, 2024.

[219] H. Yan, X. Li, X. Qiu, and B. Deng. BERT for monolingual and cross-lingual reverse dictionary. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4329–4338, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.388. URL https://aclanthology.org/2020.findings-emnlp.388.

[220] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[221] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang. Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio,*

Speech, and Language Processing, 26(3):671–681, 2018. doi: 10.1109/TASLP.2017.2788182.

[222] W. Yu, C. Zhu, Y. Fang, D. Yu, S. Wang, Y. Xu, M. Zeng, and M. Jiang. Dictbert: Enhancing language model pre-training with dictionary, 2022.

[223] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma, Y. Xue, J. Zhai, W. Chen, P. Zhang, Y. Dong, and J. Tang. Glm-130b: An open bilingual pre-trained model, 2023. URL `https://arxiv.org/abs/2210.02414`.

[224] L. Zhang, F. Qi, Z. Liu, Y. Wang, Q. Liu, and M. Sun. Multi-channel reverse dictionary model. *arXiv preprint arXiv:1912.08441*, 2019.

[225] M. R. Zhang, N. Desai, J. Bae, J. Lorraine, and J. Ba. Using large language models for hyperparameter optimization, 2022. URL `https://arxiv.org/abs/2312.04528`.

[226] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models, 2023.

[227] R. Zhu, T. Noraset, A. Liu, W. Jiang, and D. Downey. Multi-sense definition modeling using word sense decompositions, 2019.

[228] M. Zock. Word lookup as an ongoing dialogue between a user and a lexicon. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pages 484–487, 2004.

[229] M. Zock, O. Ferret, and D. Schwab. Deliberate word access: an intuition, a roadmap and some preliminary empirical results. *International Journal of Speech Technology*, 13(4):201–218, 2010.

Appendix A

Appendix A: GDEX-based Prompt

Given the word {word}, defined as {definition}, write a sentence that shows this word being used in context, in the style of a dictionary or an encyclopedia. The returned sentence should follow the following criteria:

- it should be between 10 and 25 words.
- it should appear natural.
- it must show the typical usage of the word in terms of context, syntax, phraseology and the like.
- it should be informative and helps the user with understanding the definition.
- its content should be understandable without the need for wider context.
- it should include the least number of pronouns.
- it should include the least number of non-frequent words.
- it should not have more than one capital letter or non-alpha-numeric characters.
- it should not be another definition of the target word. Do not generate more definitions or descriptions of the term.
- the target word should be presented in the main clause.
- it should contains the most typical collocates of the target word.

Here are some examples:
INPUT: asleep, in a state of sleep
OUTPUT: Sometimes we watched movies and fell asleep together.

Appendix B

Appendix B: GDEX and Readability-based Examples Evaluation Statistics

Dataset	Avg	Std	Min	Max
WN	0.39	0.17	0	0.85
CHA	0.47	0.09	0	0.76
Wiktionary	0.50	0.09	0	0.90
Wikipedia	0.48	0.07	0	0.81
Urban	0.41	0.12	0	0.93
CODWOE	0.49	0.12	0	0.90
Sci-definition	0.55	0.08	0	0.89

Table B.1: Fluency

Dataset	Avg	Std	Min	Max
WN	0.71	0.27	0	0.95
CHA	0.16	0.31	0	0.99
Wiktionary	0.47	0.42	0	0.99
Wikipedia	0.16	0.29	0	0.93
Urban	0.34	0.37	0	0.94
CODWOE	0.51	0.41	0	0.99
Sci-definition	0.35	0.41	0	0.99

Table B.2: Length penalty

Dataset	Avg	Std	Min	Max
WN	0.18	0.20	0	1
CHA	0.09	0.08	0	1
Wiktionary	0.23	0.12	0	1
Wikipedia	0.23	0.12	0	1
Urban	0.24	0.16	0	1
CODWOE	0.20	0.13	0	1
Sci-definition	0.20	0.12	0	1

Table B.3: Frequency penalty

Dataset	Avg	Std	Min	Max
WN	0.07	0.11	0	0.66
CHA	0.09	0.07	0	0.76
Wiktionary	0.09	0.07	0	0.75
Wikipedia	0.02	0.03	0	0.42
Urban	0.12	0.09	0	0.75
CODWOE	0.09	0.08	0	0.66
Sci-definition	0.05	0.051	0	0.50

Table B.4: Anaphoric penalty

Dataset	Avg	Std	Min	Max
WN	0.59	0.06	0.27	1
CHA	0.57	0.03	0.36	0.95
Wiktionary	0.54	0.04	0.20	0.94
Wikipedia	0.55	0.04	0.24	0.83
Urban	0.54	0.05	0.04	0.97
CODWOE	0.55	0.05	0.22	0.96
Sci-definition	0.57	0.03	0	0.82

Table B.5: Ambiguity

Dataset	Avg	Std	Min	Max
WN	0.98	0.15	0	1
CHA	0.84	0.36	0	1
Wiktionary	0.68	0.46	0	1
Wikipedia	0.96	0.19	0	1
Urban	0.84	0.36	0	1
CODWOE	0.79	0.40	0	1
Sci-definition	0.85	0.35	0	1

Table B.6: Main clause

Dataset	Avg	Std	Min	Max
WN	7.23	6.47	-10.48	50.090
CHA	9.12	4.07	-6.55	36.19
Wiktionary	11.81	6.37	-6.56	96.27
Wikipedia	11.30	3.62	-6.55	33.95
Urban	4.42	3.58	-12.45	52.44
Cod	9.44	6.24	-8.23	67.79
Sci	17.06	7.40	-3.01	289.29

Table B.7: Flesch–Kincaid Reading Grade Level (FKRGL)

Dataset	Avg	Std	Min	Max
WN	8.81	4.91	0.09	19.72
CHA	9.63	2.16	0.89	19.92
Wiktionary	9.33	2.81	0.09	23.30
Wikipedia	11.30	3.62	-6.55	33.95
Urban	8.62	2.87	0	21.01
Cod	8.77	3.14	0.09	20.02
Sci	12.26	2.24	0.09	50.60

Table B.8: Dale-Chall Readability (DCR)

Dataset	Avg	Std	Min	Max
WN	8.23	7.41	-17.82	57.600
CHA	9.16	4.23	-15.90	35.53
Wiktionary	10.66	5.09	-18.84	313.44
Wikipedia	11.15	3.94	-15.90	43.47
Urban	4.92	4.70	-27.66	466.16
Cod	8.98	5.17	-18.84	81.12
Sci	15.85	4.82	-10.02	442.74

Table B.9: Coleman-Liau Index (CLI)

Appendix C

Appendix C: GEAR Prompt Types

Base Prompt 1 (bp1):

Given the definition {definition}, generate a list of {k} terms defined by that definition assuming they are in {dictionary} dictionary. Only give me a list back, do not generate any other text. {dictionary} is {description}

The returned list should follow the following conditions:

- Terms should be ordered or ranked so the first term is the most related to the definition.
- In a JSON object of the form { "terms": ["term_1", "term_2", . . .] }.
- All terms should be in lowercase.

Example:

INPUT: "A piece of furniture for sitting."

OUTPUT: { "terms": ["chair", "stool", "bench", "sofa", "couch"] }

Base Prompt 2 (bp2):

Given the definition {definition}, generate a list of {k} terms defined by that definition assuming they are in {dictionary} dictionary. Only give me a list back, do not generate any other text. {dictionary} is {description}

These are some examples of definitions and terms in this dictionary: {examples}

The returned list should follow the following conditions:

- Terms should be ordered or ranked so the first term is the most related to the definition.
- In a JSON object of the form { "terms": ["term_1", "term_2", . . .] }.
- All terms should be in lowercase.

Example:

INPUT: "A piece of furniture for sitting."This compile didn't produce a PDF. This can happen if:

OUTPUT: { "terms": ["chair", "stool", "bench", "sofa", "couch"] }

Reasoning Prompt (rp):

Given the definition {definition}, generate a list of {k} terms defined by that definition assuming they are in {dictionary} dictionary. Only give me a list back, do not generate any other text.
{dictionary} is {description}
These are some examples of definitions and terms in this dictionary: {examples}
For each term, provide an example usage in a sentence that matches the style and scope of {dictionary}.
The returned list should follow the following conditions:

- Terms should be ranked, with the first term being the most related to the definition.
- All terms and examples should be in lowercase.
- Return the terms and examples in a JSON object of the form:

{ "terms": [{ "term": "term_1", "example": "example_1" }, { "term": "term_2", "example": "example_2" }, ...] }
Example:
INPUT: "A piece of furniture for sitting."
OUTPUT: { "terms": [{ "term": "chair", "example": "he sat on the chair and opened his book." }, { "term": "stool", "example": "she perched on the stool at the bar." }, { "term": "bench", "example": "they rested on the bench after their walk." }, { "term": "sofa", "example": "the family gathered on the sofa to watch TV." }, { "term": "couch", "example": "he stretched out on the couch to take a nap." }] }

Appendix D

Appendix D: GEAR Generated Terms

The following tables present samples of candidates generated by `gpt-4o-mini` for each prompt across sources in the 3D-EX dataset, and by both `gpt-4o-mini` and `Llama 3.1` for the *description* test set of Hill's dataset. The *seen* and *unseen* sets are excluded, as they are derived from WordNet.

Here, we examine sample outputs to explore how different factors affect the model's predictions. In the WN results as shown in Table D.1, the model's output varies depending on the prompt and sometimes misses the exact word type or tone intended by the dictionary. For example, for the definition *Having no limbs*, the base prompt mostly lists animals without limbs instead of the correct adjective *limbless*. Adding examples to the prompt helps the model produce better adjective terms like *limbless* and *armless*. For the definition *The relative position or standing of things or especially persons in a society*, the model's top prediction is *social status*, which is not in the gold list but is closely related, showing a broader understanding. Finally, for *Displaying luxury and furnishing gratification to the senses*, the gold terms are adjectives like *luxurious* and *voluptuous*, but the model mostly predicts nouns such as *luxury* and *opulence*, except in the reasoning prompt where adjectives like *opulent* and *sumptuous* appear. The model's terms also tend to have a stronger, more judgmental tone compared to the gold terms, possibly reflecting an overcautious or exaggerated view of luxury. This might suggest that the LLM is overcompensating by being too careful or safe in its word choices.

More generally, the style of each dictionary strongly influences the model's generated terms. For example, the Urban dataset (Table D.5) features casual, slangy language; for the definition *A social gathering when there is not enough people to call it a party*, the model suggests informal terms like *get-together* and *chill sesh*, matching the casual tone of the gold term *box social*. In the Sci-definition dataset of medical and technical terms (Table D.7), the model produces accurate, domain-specific vocabulary and related concepts. The Hei++ dataset (Table D.10) focuses on adjective-noun phrases, and the model consistently preserves this structure with terms like *prompt reply* and *swift response*. Across these datasets, prompts bp2 and rp tend to produce more formal, specialized, or creative terms, while bp1 usually generates simpler, broader, or more literal language.

Table D.11 presents a comparison of terms generated by `gpt-4o-mini` and `Llama-3` for the description set in Hill's dataset. Both models successfully identify the core concept in each definition and produce related terms that reflect the intended meaning; however, `Llama-3` tends to produce more varied and dynamic language, while `gpt-4o-mini` offers more precise and focused terms. For example, in the case of *Dirty wet brown earth found on the ground after it has rained*, both models include the key term *mud* along with similar words such as *sludge* and *silt*. However, `gpt-4o-mini` tends to generate more general terms like *soil* and *dirt*, while `Llama-3` offers some less common words such as *muck* and *ooze*. For the definition *To keep doing something when stopping was a possibility*, both models provide terms related to the term *continue*, but `gpt-4o-mini` generates nouns such as *perseverance* and *determination*, whereas `Llama-3` produces more verbs and phrases like *carry on* and *press on*. In the case of *The process where the price of things increases across a country or economy*, both identify *inflation* and related concepts. However, `Llama-3` generates a broader range of economic terms while `gpt-4o-mini` remains more focused on terms directly related to price increases.

Definition	Gold Terms	Prompt	Pred. Terms
The relative position or standing of things or especially persons in a society	[position, status]	bp1	[social status, socioeconomic status, rank, class, position]
		bp2	[social status, social class, rank, standing, hierarchy]
		rp	[social status, standing, class, rank, hierarchy]
Having no limbs	[limbless]	bp1	[worm, snake, mollusk, jellyfish, slug]
		bp2	[unarmored, limbless, defenseless, impaired, incapacitated]
		rp	[armless, legless, limbless, bodyless, trunked]
Of or relating to or characteristic of a procession	[processional]	bp1	[processional, parade, procession, pageant, cortege]
		bp2	[processional, processionary, parade, cortège, pageant]
		rp	[processional, parade, march, cortege, pageant]
Displaying luxury and furnishing gratification to the senses	[luxurious, voluptuous, epicurean, sybaritic]	bp1	[luxury, opulence, splendor, lavishness, indulgence]
		bp2	[opulence, luxury, delight, lavishness, extravagance]
		rp	[opulent, lavish, sumptuous, extravagant, luxurious]
(of flora or fauna) likely in the near future to become endangered	[threatened]	bp1	[threatened species, vulnerable species, endangered species, imperiled species, at risk species]
		bp2	[vulnerable, threatened, endangered, critically endangered, near threatened]
		rp	[threatened, vulnerable, near threatened, endangered, at risk]
Consider obligatory; request and expect	[ask, expect, require]	bp1	[demand, require, insist, urge, command]
		bp2	[demand, require, insist, urge, call for]
		rp	[demand, require, obligate, necessitate, expect]

Table D.1: Sample of generated terms in WordNet.

Definition	Gold Terms	Prompt	Pred. Terms
An agreement or treaty especially one between the Vatican and a secular government relating to matters of mutual interest	[concordat]	bp1	[concordat, treaty, agreement, pact, accord]
		bp2	[concordat, treaty, accord, agreement, pact]
		rp	[concordat, pact, treaty, agreement, accord]
A person who practises or is an expert in judo	[judoka]	bp1	[judoka, judo practitioner, judo expert, judo player, judo athlete]
		bp2	[judoka, judo practitioner, judo expert, judoist, judo athlete]
		rp	[judoka, judo expert, judo practitioner, judo champion, judo athlete]
A cubic equation or a curve described by one	[cubic]	bp1	[cubic equation, parabola, cubic function, polynomial curve, three-dimensional graph]
		bp2	[cubic equation, cubic function, polynomial equation, cubic curve, third-degree polynomial]
		rp	[cubic function, cubic polynomial, cubic curve, cubic equation, cubic root]
Denoting or relating to the part of a political administration with responsibility for putting into effect laws drawn up by the legislature	[executive]	bp1	[executive, administration, executive branch, government, executive authority]
		bp2	[executive, administration, governance, bureaucracy, execution]
		rp	[executive, administration, government, bureaucracy, cabinet]
An air-filled swelling in a plant especially a seaweed	[vesicle]	bp1	[bladder, bloat, float, air sac, pneumatophore]
		bp2	[bladder, float, air sac, pneumatocyst, aerenchyma]
		rp	[bladder, float, pneumatocyst, air sac, buoyancy chamber]
Having lost the physical, mental, or moral qualities considered normal and desirable; showing evidence of decline	[degenerate]	bp1	[decrepit, dilapidated, decayed, degenerate, worn]
		bp2	[degeneration, decay, deterioration, neglect, decline]
		rp	[decadent, dilapidated, decrepit, degraded, degenerate]

Table D.2: Sample of generated terms in CHA dataset.

Definition	Gold Terms	Prompt	Pred. Terms
City in Namur, Belgium	[andenne, beauraing, fosses-la-ville]	bp1	[namur, belgium, province of namur, walloon region, city]
		bp2	[namur, wallonia, meuse, namur castle, st. aubin's cathedral]
		rp	[namur, floreffe, ham-sur-heure-nalinnes, beaumont, église]
Harmful effects of human activity on the biophysical environment	[environmental issue]	bp1	[environmental degradation, pollution, deforestation, biodiversity loss, climate change]
		bp2	[environmental degradation, pollution, deforestation, biodiversity loss, climate change]
		rp	[environmental degradation, pollution, deforestation, climate change, habitat destruction]
Academic degree	[bachelor of business administration, bachelor of music, doctor of divinity, lector, master of laws, professional degree]	bp1	[bachelor's degree, master's degree, doctorate, associate degree, professional degree]
		bp2	[bachelor's degree, master's degree, doctorate, associate degree, professional degree]
		rp	[bachelor's degree, master's degree, phd, associate degree, doctorate]
Computer programming language	[logo, sail]	bp1	[programming language, scripting language, markup language, compiled language, interpreted language]
		bp2	[python, java, javascript, c++, ruby]
		rp	[python, javascript, java, c++, ruby]
Type of musical instrument	[electric instrument, gemshorn]	bp1	[guitar, piano, violin, trumpet, drums]
		bp2	[piano, guitar, violin, flute, drum]
		rp	[piano, guitar, violin, trumpet, flute]
Soft drink	[sparletta, tab, vimto]	bp1	[soda, pop, cola, fizzy drink, carbonated beverage]
		bp2	[soda, pop, cola, fizzy drink, carbonated beverage]
		rp	[cola, lemonade, ginger ale, root beer, sparkling water]

Table D.3: Sample of generated terms in Wikipedia.

Definition	Gold Terms	Prompt	Pred. Terms
The act of writing a computer program	[programming]	bp1	[programming, coding, software development, scripting, application development]
		bp2	[programming, coding, software development, scriptwriting, algorithm design]
		rp	[programming, coding, software development, application development, scriptwriting]
Truly, honestly	[fair dinkum, straight dinkum]	bp1	[genuinely, sincerely, authentically, veritably, frankly]
		bp2	[genuinely, actually, seriously, sincerely, veritably]
		rp	[genuinely, truthfully, honestly, sincerely, forthrightly]
A very brief vacation; a momentary respite from the cares of everyday life	[microvacation, nanobreak]	bp1	[getaway, retreat, break, escape, holiday]
		bp2	[getaway, break, escapade, retreat, recess]
		rp	[getaway, break, escape, recess, hiatus]
Having multiple targets, such that a call to the delegate triggers a call to each target	[multicast]	bp1	[multicast, broadcast, delegate, event, callback]"
		bp2	[multicast, broadcast, delegate, listener, subscriber]"
		rp	[multicast, broadcast, event aggregator, observer pattern, delegate]
A procedure used to find a particular solution to certain types of inhomogeneous ordinary differential equation	[variation of parameters, undetermined coefficients, integrating factor, particular solution, method of undetermined coefficients]	bp1	[particular solution, method of undetermined coefficients, variation of parameters, laplace transform, green's function]
		bp2	[variation of parameters, undetermined coefficients, particular integral, method of undetermined coefficients, variation method]
Based on hypothesis rather than experiment	[a priori]	bp1	[theory, hypothesis, model, conjecture, postulate]
		bp2	[hypothetical, theoretical, conjectural, speculative, supposititious]
		rp	[theoretical, hypothetical, presumed, conjectural, supposed]

Table D.4: Sample of generated terms in Wiktionary.

Definition	Gold Terms	Prompt	Pred. Terms
A social gathering when there's not enough people to entitle it a party	[box social]	bp1	[get-together, small hangout, mini gathering, casual meetup, social bump]
		bp2	[mini-gathering, not-a-party, micro-party, chill sesh, friend hang]
Being able to drive and refold a road map at the same time	[accordianated, accordionated]	bp1	[hangout, get-together, mini-gathering, low-key meet-up, casual congregation]
		bp2	[multitasking navigator, map wrangler, dual-tasking driver, roadmaster, cartographic multitasker]
		rp	[map multitasker, navitationalist, road map wizard, fold-n-drive, compass contortionist]
To understand	[underdig, feel]	bp1	[get it, vibe check, deep dive, red pill, woke]
		bp2	[get it, feel me, you dig, catch my drift, know what I mean]
		rp	[get it, catch on, dig, vibe with, grok]
The way you dress and how you present yourself to other people	[swaga]	bp1	[style, aesthetic, fit, vibe, swag]
		bp2	[swagmaster, flex god, style king, dancefloor deity, vibe guru]
		rp	[aesthetic, vibe check, fit, swag, drip]
Generally being stressed out at your computer whilst playing a game or engaging in any other online activity	[e-rage]	bp1	[rage quitting, gamer stress, tilted, pwned, noob mode]
		bp2	[rage quit, gamer stress, tilt mode, keyboard smash, online meltdown]
		rp	[gamer stress, online anxiety, click rage, screen burnout, keyboard freakout]
A class that is useless and extremely difficult for those who are not math geniuses	[algebra ii]	bp1	[math pit, number hell, calc-tastrophe, algebraic abyss, geometry graveyard]
		bp2	[math hell, calculator torture, numbers nightmare, algebra apocalypse, geometry graveyard]
		rp	[math hell, calculussive, numbrageous, geometry of despair, algebra blaster]

Table D.5: Sample of generated terms in Urban.

Definition	**Gold Terms**	**Prompt**	**Pred. Terms**
Complimentary	[complemental, free]	bp1	[complimentary, gratuitous, favorable, laudatory, commendatory]
		bp2	[praiseworthy, flattering, gratis, favorable, encomiastic]
		rp	[complimentary, flattering, praising, admiring, laudatory]
A photograph	[photo, picture]	bp1	[photo, image, snapshot, picture, developed photograph]
		bp2	[photo, picture, snapshot, image, portrait]
		rp	[photo, snapshot, picture, image, print]
To commit suicide	[rope, top]	bp1	[suicide, self-harm, self-inflicted death, voluntary death, intentional death]
		bp2	,[suicide, self-termination, self-inflicted death, voluntary death, self-killing]
		rp	[suicide, self-termination, self-inflicted death, voluntary death, self-killing]
The philosophy of using copyrights to enforce freedom of information, especially software source code	[copyleft]	bp1	[copyleft, open source, free software, software freedom, information liberation]
		bp2	[copyleft, free software, open source, software freedom, information commons]
		rp	[copyleft, free software, open source, intellectual property, information freedom]
A pair of one-way streets which carry opposing directions of traffic through gridded urban areas	[couplet]	bp1	[one-way street, dual carriageway, one-way system, two-way street, traffic corridor]
		bp2	[one-way street, traffic avenue, divergent roadway, contraflow street, grid street]
		rp	[one-way street, concurrent street, directional lane, grid street, paired thoroughfare]

Table D.6: Sample of generated terms in CODWOE dataset.

Definition	Gold Terms	Prompt	Pred. Terms
A method of seeding that involves placing seed with attention to precise spacing and depth, either by hand or mechanically, as opposed to broadcast seeding	[precision seeding]	bp1	[precision seeding, direct seeding, row planting, planting technique, mechanical seeding]
		bp2	[precision seeding, mechanical seeding, direct seeding, row seeding, transplanting]
		rp	[precision seeding, drill seeding, row planting, transplanting, direct seeding]
Bronchitis is an inflammation of the bronchial tubes, the airways that carry air to your lungs	[acute bronchitis, chronic bronchitis]	bp1	[bronchitis, bronchial tubes, airways, lungs, respiratory system]
		bp2	[bronchitis, asthma, chronic obstructive pulmonary disease, pneumonia, lung infection]
		rp	[bronchitis, bronchial tube inflammation, respiratory infection, airway obstruction, lung disease]
A crime against environmental legislation that is liable for prosecution	[environmental crime]	bp1	[environmental crime, pollution offense, illegal dumping, wildlife trafficking, hazardous waste violation]
		bp2	[environmental crime, ecological violation, green law offense, pollution offense, conservation breach]
		rp	[environmental crime, eco-crime, pollution offense, wildlife trafficking, illegal logging]
Any area of land that is lower in elevation relative to another area	[lowland]	bp1	[valley, depression, basin, glen, hollow]
		bp2	[basin, valley, depression, lowland, fjord]
		rp	[depression, valley, basin, lowland, sinkhole]
An endogenous compound that enables neurotransmission	[neurotransmitter]	bp1	[neurotransmitter, serotonin, dopamine, glutamate, acetylcholine]
		bp2	[neurotransmitter, acetylcholine, dopamine, serotonin, norepinephrine]
		rp	[neurotransmitter, serotonin, acetylcholine, glutamate, gamma-aminobutyric acid]

Table D.7: Sample of generated terms in Sci-definition dataset.

Definition	Gold Terms	Prompt	Pred. Terms
The state of being a landlord	[landlordism]	bp1	[landlordship, property ownership, landlording, real estate management, rental property management]
		bp2	[landlordship, landholding, property ownership, renting, leaseholding]
		rp	[landlordship, landlording, property owner, lessor, rental proprietor]
Blindness	[ablepsy, cecity]	bp1	[blindness, visual impairment, sightlessness, hypopsia, amaurosis]
		bp2	[blindness, amaurosis, insensibility, unseeing, lack of vision]
		rp	[blindness, visual impairment, anopsia, amaurosis, legal blindness]
An instrument consisting in part of a differential thermometer	[aethrioscope]	bp1	[thermometer, differential thermometer, max-min thermometer, manometer, barometer]
		bp2	[thermometer, differential thermometer, hygrometer, barometer, manometer]
		rp	[manometer, barometer, thermograph, calorimeter, potentiometer]
A salt of sulphophosphorous acid	[sulphophosphite]	bp1	[sulphophosphate, sulphophosphoric acid, sulphophosphate of lime, sulphophosphate of soda, sulphophosphite]
		bp2	[sodium thiosulphate, potassium thiosulphate, calcium thiosulphate, ammonium thiosulphate, magnesium thiosulphate]
		rp	[sulphophosphate, sulphophosphite, calcium sulphophosphate, sodium sulphophosphate, magnesium sulphophosphate]
Toward the southeast	[southeaster, southeastward, southeastwardly]	bp1	[southeast, southeastward, southeastwardly, southeasterly, southeaster]
		bp2	[southeast, southeastward, southeasterly, southeastern, southeastwardly]
		rp	[southeast, southeasterly, southeastward, southeastwardly, southeast-bound]
A judicial opinion expressed by judges on points that do not necessarily arise in the case, and are not involved in it	[dictum]	bp1	[obiter dictum, dictum, pertinent statement, extrajudicial remark, judicial commentary]
		bp2	[obiter dictum, dictum, advisory opinion, extrajudicial opinion, res ipsa loquitur]
		rp	[dictum, obiter dictum, advisory opinion, extra judicial, non sequitur]

Table D.8: Sample of generated terms in Webster's Unbridged.

Definition	Gold Terms	Prompt	Pred. Terms
A community of people smaller than a village	[crossroads, hamlet]	bp1	[hamlet, settlement, borough, neighborhood, community]
		bp2	[hamlet, settlement, borough, community, locality]
		rp	[hamlet, settlement, borough, neighborhood, colony]
A tentative insight into the natural world; a concept that is not yet verified but that if true would explain certain facts or phenomena	[hypothesis, possibility, theory]	bp1	[hypothesis, theory, model, speculation, postulate]
		bp2	[hypothesis, theory, postulate, conjecture, speculation]
		rp	[hypothesis, theory, postulate, conjecture, speculation]
The science which treats of wealth, its production, distribution, etc.; political economy	[economics]	bp1	[economics, political economy, finance, market theory, resource allocation]
		bp2	[economics, political economy, wealth, distribution, production]
		rp	[economics, political economy, economy, wealth, distribution]
Small dark geese that breed in the north and migrate southward	[brant, brent]	bp1	[brant, barnacle goose, canada goose, white-fronted goose, lesser snow goose]
		bp2	[brent goose, barnacle goose, cackling goose, canada goose, little blue goose]
		rp	[brant, barnacle goose, white-fronted goose, cackling goose, greater white-fronted goose]
A swimming stroke in which a person lies face down in the water and extends the arms in front of the head, then sweeps them both back laterally under the surface of the water while performing a frog kick	[breaststroke]	bp1	[breaststroke, freestyle, butterfly, backstroke, sidestroke]
		bp2	[breaststroke, sweep stroke, frog stroke, horizontal stroke, water stroke]
		rp	[breaststroke, frog stroke, front crawl, swimming technique, water fitness]
The introduction of animals or plants to places where they flourish but are not indigenous	[naturalization]	bp1	[invasive species, non-native species, introduced species, alien species, exotic species]
		bp2	[invasive species, introduced species, non-native species, exotic species, alien species]
		rp	[invasive species, exotic species, non-native species, alien species, introduced species]

Table D.9: Generated terms in MultiRD dataset.

Definition	Gold Terms	Prompt	Pred. Terms
A line that is drawn with no interruptions between its ends	[continuous line]	bp1	[straight line, line segment, continuous line, uninterrupted line, infinite line]
		bp2	[straight line, continuous line, uninterrupted line, closed line, open line]
		rp	[line segment, straight line, connecting line, continuous line, uninterrupted line]
A necktie that is very noticeable	[conspicuous necktie]	bp1	[loud necktie, flashy necktie, colorful necktie, patterned necktie, bold necktie]
		bp2	[loud tie, flashy necktie, bold tie, striking cravat, eye-catching neckwear]
		rp	[loud tie, statement tie, bold necktie, flashy tie, eye-catching tie]
A reply that is made quickly	[prompt reply]	bp1	[quick response, instant reply, rapid answer, swift feedback, prompt remark]
		bp2	[quick reply, instant response, prompt answer, rapid feedback, swift reaction]
		rp	[quick reply, swift response, rapid comeback, prompt answer, immediate feedback]
An unpleasant noise that continues without interruption	[incessant noise]	bp1	[din, cacophony, clamor, racket, commotion]
		bp2	[constant noise, persistent racket, ceaseless clamor, unending din, relentless discord]
		rp	[din, racket, clamor, cacophony, uproar]
A change which reforms the essential qualities of something	[fundamental change]	bp1	[transformation, reformation, metamorphosis, revamping, overhaul]
		bp2	[transformative change, revolutionary alteration, qualitative reform, fundamental modification, essential revision]
		rp	[transformation, metamorphosis, reformation, conversion, revitalization]
The sun shining during the hours of maximum light	[bright sun]	bp1	[daylight, sunlight, bright sunlight, high noon, golden hour]
		bp2	[daylight, sunlight, bright hours, solar peak, golden hour]
		rp	[daylight, sunshine, midday light, solar glow, bright hours]

Table D.10: Sample of generated terms in Hei++ dataset.

Definition	Gold Term	Prompt	Pred. Terms
Dirty wet brown earth found on the ground after it has rained	mud	bp1	[mud, sludge, silt, soil, clay]
		bp2	[mud, sludge, dirt, mire, silt]
		rp	[mud, sludge, dirt, clay, silt]
To keep doing something when stopping was a possibility	continue	bp1	[perseverance, persistence, tenacity, determination, continuation]
		bp2	[persist, persevere, continue, endure, sustain]
		rp	[persist, continue, persevere, carry on, hang on]
The process where the price of things increases across a country or economy	inflation	bp1	[inflation, price increase, economic inflation, cost of living increase, hyperinflation]
		bp2	[inflation, price increase, cost of living, economic inflation, hyperinflation]
		rp	[inflation, price hike, cost of living increase, economic expansion, market volatility]

(a) Terms generated by `gpt-4o-mini`.

Definition	Gold Term	Prompt	Pred. Terms
Dirty wet brown earth found on the ground after it has rained	mud	bp1	[mud, clay, silt, sludge, mire]
		bp2	[mud, muck, dirt, clay, silt]
		rp	[mud, clay, mire, ooze, silt]
To keep doing something when stopping was a possibility	continue	bp1	[persevere, persist, continue, prevail, endure]
		bp2	[persevere, persist, continue, carry on, proceed]
		rp	[persevere, persist, keep at, press on, carry on]
The process where the price of things increases across a country or economy	inflation	bp1	[inflation, price hike, hyperinflation, stagflation, deflation]
		bp2	[inflation, price rise, cost increase, economy surge, expansion inflation]
		rp	[inflation, price hike, cost of living, price increase, economic growth]

(b) Terms generated by `Llama-3`.

Table D.11: Sample of generated terms in Hill's dataset (Description).