2022

Probing with Noise: Unpicking the Warp and Weft of Taxonomic and Thematic Meaning Representations in Static and Contextual Embeddings

Filip Klubička
Technological University Dublin, fklubicka@gmail.com

Funder: Science Foundation Ireland

Probing with Noise: Unpicking the Warp and Weft of Taxonomic and Thematic Meaning Representations in Static and Contextual Embeddings

Filip Klubička

Supervisor: Prof. John D. Kelleher

School of Computer Science

Technological University Dublin

ADAPT Centre

Thesis submitted for the degree of

Doctor of Philosophy

October 2022

I would like to dedicate this thesis to every single person who was there with me in this long and character-defining journey, even if only for a brief moment.

John D. Kelleher, as my supervisor you have been a tremendous support during this process. You have always been kind, patient and understanding, and have inspired me to grow and develop into the researcher I am today. Your encouragement has helped me persevere even during the peak of the COVID lockdowns; you deserve so much credit for how you managed to keep us all going while the world struggled, and for that I owe you a special debt of gratitude. Any PhD student would be lucky to have you as their supervisor.

I am also grateful to all my dear colleagues from TU Dublin and the ADAPT Centre, the fellow PhD students and post-docs I have met along the way. You are all such brilliant researchers and wonderful collaborators, and I have learned so much from you while treading this path. I am grateful for all the research support, networking events and commiseration opportunities you have provided; you have given me a sense of belonging to the wider scientific community, a luxury that is so often needed but not given to many. Abigail, Abhijit, Alfredo, Annika, Elizabeth, Esra, Giancarlo, Maja, Pallavi, Pierre, Que, Senja, Sheila, Teresa, Vasudevan and everyone else, I cherish the friendship and memory of working with you.

A special thanks to my friends outside of academia, also known as Ireland's premiere vocal band, Ardú. Whenever I would hit a wall with my research, I knew I could always count on our music to lift me up, to raise and elevate. You have given me an escape and a relief that is often in short supply during such an arduous project. Yet equally, you have taught me about discipline and shown me that it takes more than just talent to truly excel at something. You have given me the best housemates I could have ever asked for, the chance to sing in some unbelievable venues and have enriched my life with beautiful harmonies, concert tours and major craic. Members have changed over the years, but you have all left a profound impact on me and the value you have given my life is immeasurable. Leanne, Ciarán, Vicky, Tristan, Laura, Adam, Mika, Brian and Nick, thank you for feeding my soul.

The time commitments and physical distance required of a PhD, topped off by the isolationist pandemic years, have inevitably extracted a toll on many old relationships. So to all the friends from back home, to the wonderful community I left behind and do my best to stay in touch with, I wish to extend a heartfelt thank you. Even if we have drifted apart, you will always be the friends who had a hand in shaping me into the person who would embark upon this ridiculous undertaking in the first place. Petra, Hana, David, Filip, Petra, Vid, Deana, Tomica, Ivana, Dora, Sandra, Dajana, Ozana, Vlatka, Mira, Domagoj, Gregor, Andrej, Matea, Martina, Petra, Katarina, Branimir, Ivana, Ana, Davor, Matej, Andrija, Katja, Tina, Maja, Karla, Rahela, Petar, Luka, Lobel, Marina, Vinko, Tena, Renato and Selmir, to name but a few, you will always live rent-free in my heart and I will be a better person for it.

I would also like to express my deepest affection and appreciation to my family. To my mother, who gave me more love than any reasonable person could bear, who always believed in me and nurtured my curiosity and ambition, and continues to support my journey, wherever it may take me. To my sister, who is the kindest, warmest and most selfless person I know, you are the best sister, the best friend and the best life companion anyone could ask for. To my father, who devastatingly passed away three years ago and never got to witness this thesis being completed. And to my wonderful extended family who always have a second home for me, be that in Ludbreg or Betina. Throughout my life you have given me so much love and support, and made me into the human I am today. Thank you all, I love you so very much.

Finally, I want to thank Daniel. Who knew that I would find the love of my life within two weeks of moving to Ireland and starting this PhD? You have a seemingly infinite arsenal of caring smiles, loving hugs and comfortable shoulders to cry on. Your patience, compassion and understanding, as well as your kindness, support and generosity cannot be overstated. I am absolutely certain that I would not have completed this thesis had you not been there with me every step of the way. I could not have asked for a better partner. You are my favourite person. Thank you for being mine. I love you, times positive infinity, enter.

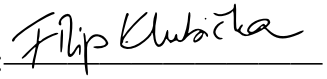
Declaration

I certify that this thesis which I now submit for examination for the award of PhD, is entirely my own work and has not been taken from the work of others, save as to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for graduate study by research of the Technological University Dublin (TU Dublin) and has not been submitted in whole or in part for another award in any other third level institution.

The work reported on in this thesis conforms to the principles and requirements of the TU Dublin's guidelines for ethics in research.

TU Dublin has permission to keep, lend or copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature: 

Date: 18 August 2022

Acknowledgements

This research was conducted with the financial support of Science Foundation Ireland under Grant Agreements No. 13/RC/2106 and 13/RC/2106_P2 at the ADAPT SFI Research Centre at Technological University Dublin. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme, and is co-funded under the European Regional Development Fund.

Abstract

The semantic relatedness of words has two key dimensions: it can be based on taxonomic information or thematic, co-occurrence-based information. These are captured by different language resources—taxonomies and natural corpora—from which we can build different computational meaning representations that are able to reflect these relationships. Vector representations are arguably the most popular meaning representations in NLP, encoding information in a shared multidimensional semantic space and allowing for distances between points to reflect relatedness between items that populate the space. Improving our understanding of how different types of linguistic information are encoded in vector space can provide valuable insights to the field of model interpretability and can further our understanding of different encoder architectures.

Alongside vector dimensions, we argue that information can be encoded in more implicit ways and hypothesise that it is possible for the vector magnitude—the norm—to also carry linguistic information. We develop a method to test this hypothesis and provide a systematic exploration of the role of the vector norm in encoding the different axes of semantic relatedness across a variety of vector representations, including taxonomic, thematic, static and contextual embeddings.

The method is an extension of the standard probing framework and allows for relative intrinsic interpretations of probing results. It relies on introducing targeted noise that ablates information encoded in embeddings and is grounded by solid baselines and confidence intervals. We call the method *probing with noise* and test the method at both the word and

sentence level, on a host of established linguistic probing tasks, as well as two new semantic probing tasks: hypernymy and idiomatic usage detection.

Our experiments show that the method is able to provide geometric insights into embeddings and can demonstrate whether the norm encodes the linguistic information being probed for. This confirms the existence of separate information containers in English word2vec, GloVe and BERT embeddings. The experiments and complementary analyses show that different encoders encode different kinds of linguistic information in the norm: taxonomic vectors store hypernym-hyponym information in the norm, while non-taxonomic vectors do not. Meanwhile, non-taxonomic GloVe embeddings encode syntactic and sentence length information in the vector norm, while the contextual BERT encodes contextual incongruity.

Our method can thus reveal where in the embeddings certain information is contained. Furthermore, it can be supplemented by an array of post-hoc analyses that reveal how information is encoded as well, thus offering valuable structural and geometric insights into the different types of embeddings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational semantics studies how to automate the process of constructing and reasoning with meaning representations of natural language expressions, be it words, phrases, sentences or even entire documents. It consequently plays an important role in computational linguistics as well as the discipline of natural language processing (NLP). One of the most popular and successful ways of creating meaning representations is to train a neural network that produces distributed representations called *embeddings*—vector representations of meaning embedded in a shared multidimensional semantic space.

In the past decade there has been an abundance of work that utilises neural networks for learning meaning representations for NLP (for example Mikolov et al. (2013a,b); Socher et al. (2013); Kalchbrenner et al. (2014); Kim (2014), to name but a few). These types of representations are automatically learned from a natural language corpus and are able to simultaneously encode multiple linguistic features of words. Moreover, the development of techniques such as Skip-Thought Vectors (Kiros et al., 2015) and Sent2vec (Pagliardini et al., 2018) have yielded approaches to learn distributed representations of sentences in an unsupervised manner. In the latter part of the decade, the landscape of the field has been terraformed with the release of the so-called "Muppet" models—the LSTM-based ELMo (Peters et al., 2018b) and transformer-based BERT (Devlin et al., 2018)—which were able

to generate contextualised embeddings, thus addressing the problem of polysemy. These models and their derivatives have rapidly surpassed the state of the art in all popular NLP tasks and, in doing so, have marked a new era in NLP.

Concurrently, an important discussion began permeating the public discourse on AI, namely the issue of AI ethics and, more specifically, explainable and interpretable AI (Whittlestone et al., 2019). Due to the non-transparent, or rather, human-uninterpretable way that neural networks build representations and make decisions, a subfield of explainable AI has begun to emerge across all AI disciplines, including NLP. A series of workshops started in 2018 called *BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (Alishahi et al., 2019) showcases NLP researchers' efforts to better understand the inner workings of neural network models, as they develop methods to more precisely pinpoint what these systems encode (tentatively, "learn" and "know") in terms of human-interpretable information. These efforts are aimed at various applications such as text classification (Jacovi et al., 2018), machine translation (Stahlberg et al., 2018), computational reasoning (Sommerauer et al., 2019) and many others. Interpretability efforts have also gripped the field of computational semantics, with a focus on better understanding embedding models and distributed meaning representations.

To this end the notion of *probing* (Ettinger et al., 2016; Veldhoen et al., 2016; Adi et al., 2017) has gained considerable traction in the area of interpretability of NLP models. Probing is used to analyse an embedding model's encoding of linguistic information: the core idea is that, by using embeddings produced by a pretrained embedding model as the sole input for a machine learning classifier (which in this case is called the *probe*) which is trained to predict a linguistic task, we can consider the probe's performance on the task as a proxy for assessing the extent of task-relevant linguistic knowledge the embedding model encodes in its embeddings. To give a concrete example, if we can train a machine learning model to predict whether a sentence is in the active or passive voice based only on the sentence's

embedding, this provides evidence that the embedding model is encoding voice information somewhere within the embeddings it generates. In other words, the underlying assumption is that, if a probe is able to successfully classify candidates, then the probed information must be contained in the embeddings themselves. It is particularly interesting to use probing to study linguistic properties that are encoded by embedding models which have not been explicitly designed to encode those linguistic properties, thus revealing emergent structures in embeddings. In theory, probing can be used to assess any property of language contained in a linguistic segment (word, phrase, sentence) that can be expected to be encoded by an embedding model, and has been used to probe for linguistic properties such as word order and sentence length, morphology, syntax, and to a degree even semantics and discourse structure. As such, the probing framework will form the methodological backbone of this thesis.

1.1 Research Questions and Proposed Research

Though it is still in its early stages, research on probing is rapidly developing. While its potential for application is broad, there are many NLP tasks that have not yet been explored with the probing framework. Specifically, it seems the majority of impactful probing work focuses on analysing syntactic properties encoded in language representations, yet the rich and complex field of semantics is comparably underrepresented. One particular semantic problem that has not been explored at all in the context of probing is the distinction between the *taxonomic* and *thematic* dimensions of semantic relatedness (Kacmajor and Kelleher, 2019): words or concepts which belong to a common taxonomic category share properties or functions, and such relationships are commonly reflected in knowledge-engineered resources such as ontologies or taxonomies. On the other hand, thematic relations exist by virtue of co-occurrence in a linguistic context where the relatedness is specifically formed between concepts performing complementary roles in a common event or theme. Modelling both

kinds of relationships is important for building AI with comprehensive natural language understanding abilities, however, by default, the vast majority of pretrained language models are trained solely on natural language corpora. This means that they mainly encode thematic relations, even though both types of information can be encoded by language representations. Consequently, most probing work is applied to thematic embeddings, while taxonomic embeddings remain unexplored. We wish to foreground this distinction and use the probing framework to study and compare the different types of representations, applied to two newly developed semantic probing tasks.

While using the probing framework to peek into language representations and uncover the encoding of specific types of information is an invaluable tool for the area of model interpretability, at its core the insights provided by the typical probing pipeline are somewhat limited, simply revealing whether the relevant information is contained within a language representation. Yet it would be of great interest to take the investigation further and examine the structural and geometric properties of language encodings. For example, one aspect of embeddings that has not received much attention is the contribution of the vector norm to encoding certain linguistic information. Further developing the probing methodology and adapting it would allow us to identify where and how exactly the relevant information is encoded within a representation and what the role of the vector norm is in storing this information.

Generally, our goal in this thesis is to learn more about how different types of linguistic information are encoded in embeddings. Explicitly, our overarching research questions are:

- How are different types of linguistic information encoded in embeddings?
- Is the vector norm of embeddings capable of encoding certain linguistic properties?
- What is the interaction between different types of embeddings and the way they encode linguistic properties?

In order to answer these questions and obtain geometric insights into how embeddings store linguistic information, we require a probing method that accounts for the role of the vector norm in encoding information. This means the method needs to be able to provide an intrinsic evaluation of an individual embedding representation, while simultaneously allowing for a relative interpretation of results in order to isolate the role of the vector norm relative to the vector dimensions. However, the typical probing pipeline is not designed to provide this type of insight, as it can only tell us how well an embedding encodes some type of linguistic information when compared to another embedding model.

We thus propose an extension to the existing probing framework: first we apply the standard probing pipeline to a given task by training a probing classifier to predict linguistic features based solely on embeddings as input. We then add a further step and introduce targeted random noise into the embeddings, followed by retraining the classifier. This allows us to examine how the added noise impacts the probe's evaluation scores—if the probe's performance drops, this means informative features have been removed from the embedding.

Essentially, we examine whether the noise disrupted the information in the embedding being tested, and the right application of noise enables us to determine which embedding component the relevant information is encoded in, by ablating that component's information. In turn, this can inform our understanding of how certain linguistic properties are encoded in vector space: while the standard probing framework enables us to examine how well a vector representation encodes some type of linguistic information, our extended method enables us to examine where in the embedding this information is encoded. This allows us to perform an intrinsic evaluation of a single encoder and provides geometric insights into the encoder's embeddings. We call the method *probing with noise* and in this thesis we demonstrate its applicability to taxonomic and thematic embeddings, as well as contextual and static encoders, by using it to intrinsically evaluate English SGNS, GloVe and BERT

embeddings on ten established linguistic probing tasks, as well as two newly developed semantic probing tasks that represent taxonomic and thematic aspects of meaning.

1.2 Contributions

The major research contributions arising from the PhD research as presented in this thesis are as follows:

1. a methodological extension of the probing framework: *probing with noise*, which provides structural insights into embeddings
2. an array of experiments validating the *probing with noise* method and demonstrating its generalisability to a range of encoders and probing tasks
3. the identification of a gap in the probing literature regarding a lack of study of semantic tasks, and the consecutive development of two new semantic probing tasks: hypernym-hyponym and idiomatic usage prediction
4. the development and publication of a large set of taxonomic word embeddings and pseudo-corpora
5. a systematic exploration of the importance of the vector norm in encoding different types of linguistic phenomena in different embedding models, which shows that the norm is able to encode different types of linguistic information, with the particular information being dependant on the embedding model
6. a comparative analysis of taxonomic and thematic embeddings that reveals only taxonomic embeddings carry taxonomic information in their norm, indicating that the role of the norm can be determined by the embedding training data, i.e. the underlying distribution, rather than the model architecture

7. a comparative analysis of contextual and static embeddings that reveals significant structural differences in their respective vector spaces and shows that contextual embeddings partially encode contextual incongruity information in their vector norm.

1.2.1 Other Contributions

During the course of this PhD program, additional research contributions, including a number of accompanying publications, have been made on various topics on embeddings as well as other areas in NLP. These will not not be presented in any of the chapters as they fall outside the scope of the main strand of the research presented in the thesis. Some initial publications describe work that was carried over from past projects, like the work on a reference corpus of Croatian (Ljubešić et al., 2018), research on manual evaluation of neural machine translation systems (Klubička et al., 2018b) completed during a master's program, as well as research on hate speech detection on Twitter conducted during an Erasmus internship (Klubička and Fernández, 2018).

During the course of the PhD we first explored the practical implications of using certain evaluation metrics for selection of machine learning models (Klubička et al., 2018a), with a case study on the task of idiom token identification, which led us into the space of semantics and figurative meaning. We initially participated on a shared task on hypernym discovery (Maldonado and Klubička, 2018), which led us to exploring the applications of word embeddings to encode taxonomic knowledge. We invested considerable effort developing experiments that allowed us to gain a more in depth understanding the WordNet random walk as an algorithm for generating a pseudo-corpus which is used to train word embeddings that encode taxonomic information. During the course of our research we performed extensive experiments trying to answer the question of how large a pseudo-corpus should be to encode useful amounts of taxonomic knowledge when combined with thematic

embeddings. We have found that there is a sweet spot that can be struck in the balance between taxonomic and thematic information (Maldonado et al., 2019).

On the thematic side of things, we have done additional research on using BERT to perform idiom token classification based on an existing verb-noun multiword expression dataset. One of the main contributions of the paper is our use of the game theory concept of Shapley Values to rank the usefulness of individual idiomatic expressions for model training and using this ranking to analyse the type of information that the model finds useful in making a prediction in a typical probing setting. We find that a combination of idiom-intrinsic and topic-based properties contribute to an expression's usefulness in idiom token identification. We also show that BERT outperforms Skip-Thought sentence representations, which held the previous state of the art on that particular dataset (Nedumpozhimana et al., 2022).

It is also worth noting that during the latter half the PhD programme, a number of contributions have been made towards collecting, cleaning and processing Croatian-English parallel corpora for the PRINCIPLE project (Way et al., 2020)[1]. The project's main aim was to identify, collect and process high-quality language resources for four under-resourced European languages with the aim of developing machine translation systems for these languages. A large amount of parallel data was collected, with a focus on the eProcurement and eJustice domains. Most of the collected corpora for Croatian have been published and are freely available on the ELRC-SHARE repository[2] (Klubička et al., 2022).

Here we provide a full, chronological list of work published during the course of the programme:

- **2018**

    1. **Filip Klubička**, Antonio Toral, Victor Manuel Sánchez-Cartagena. Quantitative fine-grained human evaluation of machine translation systems: a case study on English to Croatian. *Machine Translation*. 32, 195–215. https://doi.org/10.1007/s10590-018-9214-x

[1]PRINCIPLE stands for Providing Resources in Irish, Norwegian, Croatian and Icelandic for Purposes of Language Engineering. More information can be found here: https://principleproject.eu

[2]The resources can be accessed here: https://elrc-share.eu/repository/search/?q=&selected_facets=projectFilter_exact%3APRINCIPLE%20-%20Evaluated

2. **Filip Klubička**, Giancarlo D. Salton, John D. Kelleher. Is it worth it? Budget-related evaluation metrics for model selection. *Proceedings of the 11th International Conference on Language Resources and Evaluation.* ELRA. 2014-2021.
3. **Filip Klubička**, Raquel Fernández. Examining a hate speech corpus for hate speech detection and popularity prediction. *Proceedings of 4REAL: Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language.* 16-23.
4. Alfredo Maldonado, **Filip Klubička**. ADAPT at SemEval-2018 Task 9: Skip-Gram Word Embeddings for Unsupervised Hypernym Discovery in Specialised Corpora. *Proceedings of The 12th International Workshop on Semantic Evaluation.* 924-927.
5. Nikola Ljubešić, Željko Agić, **Filip Klubička**, Vuk Batanović, Tomaž Erjavec. hr500k–A Reference Training Corpus of Croatian. *Language Technologies and Digital Humanities Conference (JT-DH 2018).* 154-161.

- **2019**

1. Alfredo Maldonado, **Filip Klubička**, John D. Kelleher. Size matters: The impact of training size in taxonomically-enriched word embeddings. *Open Computer Science.* 9:252-267. `https://doi.org/10.1515/comp-2019-0009`
2. **Filip Klubička**, Alfredo Maldonado, Abhijit Mahalunkar, John D. Kelleher. 2019. Synthetic, yet natural: Properties of WordNet random walk corpora and the impact of rare words on embedding performance. *Proceedings of the 10th Global Wordnet Conference 2019*. 140-150.

- **2020**

1. **Filip Klubička**, Alfredo Maldonado, Abhijit Mahalunkar, John D. Kelleher. English WordNet Random Walk Pseudo-Corpora. *Proceedings of The 12th Language Resources and Evaluation Conference*. ELRA. 4893-4902.

- **2022**

1. Vasudevan Nedumpozhimana, **Filip Klubička**, John D. Kelleher. Shapley Idioms: Analysing BERT Sentence Embeddings for General Idiom Token Identification. *Frontiers In Artificial Intelligence*.
   `https://doi.org/10.3389/frai.2022.813967`
2. **Filip Klubička**, Lorena Kasunić, Danijel Blazsetin, Petra Bago. Challenges of Building Domain-Specific Parallel Corpora from Public Administration Documents. *Proceedings of the 15th Workshop on Building and Using Comparable Corpora (BUCC 2022) @LREC2022*. ELRA. 50–55.
3. Petra Bago, Sheila Castilho, Edoardo Celeste, Jane Dunne, Federico Gaspari, Níels Rúnar Gíslason, Andre Kåsen, **Filip Klubička**, Gauti Kristmannsson, Helen McHugh, Róisín Moran, Órla Ní Loinsigh, Jon Arild Olsen, Carla Parra Escartín, Akshai Ramesh, Natalia Resende, Páraic Sheridan, Andy Way. Sharing high-quality language resources in the legal domain to develop neural machine translation for less-resourced European languages: best practices, challenges and applications. *Special Issue of the Journal of Language and Law*. (*Awaiting Publication*).

Most of the work presented in this thesis currently remains unpublished, however Chapter 4 is based on research that has been published in relevant venues, presenting work from two first-author papers (Klubička et al., 2019; Klubička et al., 2020). While the work presented in other chapters is not based on any currently published work, we do acknowledge a number of collaborative publications that are topically related to Chapter 4 (Maldonado et al., 2019), Chapter 5 (Maldonado and Klubička, 2018) and Chapter 6 (Nedumpozhimana et al., 2022). However, the work presented in these papers does not contribute significantly to their respective chapters as the findings are tangentially related to the work presented in the thesis. We only mention them in the relevant related work sections, as their results otherwise fall out of scope.

1.3 Thesis Summary and Structure

- Chapter 1 provides a general introduction to the topics studied in this thesis, as well as the research questions that motivate the work. It also outlines all research contributions made by the author during the course of the PhD programme.

- Chapter 2 provides a comprehensive literature overview of the three core facets of this thesis: (1) *semantics*, (2) *embeddings* and (3) *probing*, introducing the foundational concepts that will be studied in the thesis.

- Chapter 3 describes the proposed *probing with noise* method in detail. It introduces the concept of *information containers* which motivates the exploration of the different kinds of noising functions that can be used to study the structural properties of embeddings.

- Chapter 4 describes the creation, validation and evaluation of the taxonomic embeddings to which our method will be applied.

- Chapter 5 introduces the hypernym-hyponym prediction probing task and the first batch of *probing with noise* experiments, applied to taxonomic and thematic embeddings. It contains descriptions of the dataset, models, evaluation metrics and results. It also presents supplementary post hoc experiments that provide additional insights into taxonomic embeddings.

- Chapter 6 introduces the idiomatic usage probing task and the thematic batch of *probing with noise* experiments, applied to contextual and static embeddings. It contains descriptions of the dataset, models, evaluation metrics and results, with a detailed elaboration on the motivation for the choice of train and test data split. It also discusses the limitations of the dataset used in the experiments.

- Chapter 7 presents a large suite of experiments applying the *probing with noise* method to ten established probing task datasets that test for a variety of linguistic information, on contextual and static embeddings. It also includes extensive supplementary post hoc analyses and experiments that provide further structural insights into the embeddings.

- Chapter 8 contains a synthesis of all the results and develops a discussion around the experimental findings. It also discusses possible limitations and fruitful avenues for future work.

- Chapter 9 summarises the findings and contributions made by the work.

Chapter 2

Background

The work presented in this thesis lies at the intersection of three broad topics: *semantics*, *embeddings* and *probing*. These topics permeate the text and will be making appearances in most chapters, so rather than introducing them as required on a per-chapter basis, here we provide a dedicated introduction to the general background knowledge that forms the foundation of the thesis, specify the subfields that we will inhabit and introduce relevant concepts and models that will be referenced throughout the thesis. In addition to the literature presented in this chapter, some chapters will also contain a more fine-grained related work section that discusses relevant work relating to the specific topic studied in that particular chapter.

2.1 Semantics

In its broadest sense, the linguistic domain of semantics is concerned with studying meaning. It is a rich field with a number of dominant and often competing theories, but one of the crucial questions which unites the different approaches is that of the relationship between form and meaning. Hence, in a narrower sense, semantics is concerned with the inherent meaning of linguistic structures, such as words and sentences, as linguistic expressions in

and of themselves. This is distinguished from meaning as studied in pragmatics, which is concerned with those aspects of meaning that derive from the way in which words and sentences are used (Kroeger, 2019).

Two of the most prominent issues in the field of semantics are those of lexical semantics—studying the nature of the meaning of words—and compositional semantics—studying how smaller parts, like words, combine and interact to form the meaning of larger expressions, such as phrases or sentences (Bender and Lascarides, 2019). In this thesis we will touch upon both lexical and compositional semantics, as we will be dealing with modelling the meaning of, and relationships between, words, multi-word expressions and sentences. We will do so studying examples of different dimensions of semantic relatedness, in large part using a distributional semantics lens.

Distributional semantics is founded on the distributional hypothesis (Harris, 1954; Firth, 1957), which broadly states that words which occur in the same contexts tend to have similar meanings. Based on this notion, the primary focus of distributional semantics is to develop and study theories and methods for quantifying and categorising semantic similarities between linguistic items based on their distributional properties in large samples of language data (Goldberg, 2017). In other words, its goal is to identify words/phrases/sentences that are similar to each other. However, given that semantic similarity encompasses a variety of different lexico-semantic and topical relations (Weeds et al., 2004), this raises the question of what kind of similarity is being measured, represented and ultimately evaluated in the distributional semantics literature. In fact, Kacmajor and Kelleher (2019) have found that related work on semantic relatedness and similarity often does not specify what kind of similarity is being modelled or evaluated.

While semantic relatedness is often treated as a single concept in the literature on lexical semantics, in reality there are at least two key dimensions of semantic relationships between words or concepts: **taxonomic** and **non-taxonomic**. Taxonomic relations are based on a

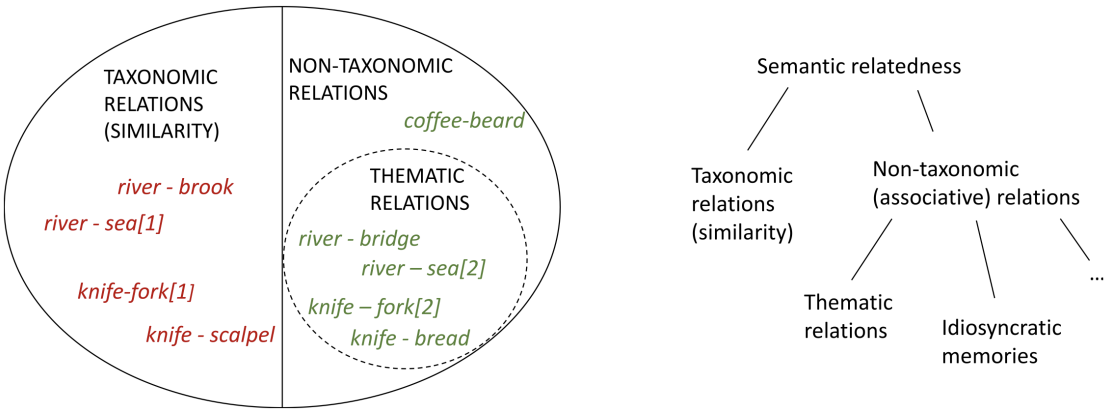


Figure 2.1 Subsets of semantic relatedness. Pairs marked [1] and [2] are examples of the same concept pairs being linked by two different relatedness types. Image originally published by Kacmajor and Kelleher (2019) as Figure 1, licensed under CC BY 4.0.

comparison of the concepts' features, meaning that concepts which belong to a common taxonomic category share properties or functions (consider *table* and *desk*). On the other hand, non-taxonomic relations exist by virtue of co-occurrence of concepts in any sort of context, for example temporal, spatial or linguistic context. An example of this would be **thematic relations** (Lin and Murphy, 2001), where the relatedness is specifically formed between concepts performing complementary roles in a common event or theme, which often implies having different features and functions which are complementary (compare *table* and *chair*). In the domain of distributional semantics, thematic relations can be considered to describe relationships between words that frequently co-occur in the same linguistic context—in the same sentence, for example. Kacmajor and Kelleher (2019) have explored this distinction between taxonomic and thematic relations in depth and have found that when "similarity" is used in the distributional semantics literature it most often refers to taxonomic similarity. They provide an informative illustration of the distinct similarity categories, as shown in Figure 2.1, and highlight the importance of this distinction, arguing that the ability to differentiate between taxonomic and thematic relations can lead to enhanced statistical language models. Each type of relation has the ability to contribute in different ways: taxonomic relations indicate which words can be replaced by other words, while thematic

relations express high-probability co-occurrences and thus help in tasks such as language modelling.

In this sense, the concepts of taxonomic and thematic relations roughly correspond to the Saussurean concepts of paradigmatic and syntagmatic relations between linguistic elements (De Saussure, 2011). **Paradigmatic** relations can be conceived as vertical, as they pertain to a relationship among linguistic elements that can substitute for each other in a given context. Given an example sentence such as *The Sun is shining*, this is the relationship of *Sun* to other nouns, such as *Moon*, *star*, or *light*, that could substitute for it in that sentence. On the other hand, **syntagmatic** relations can be conceived as horizontal, as they pertain to relationships among linguistic elements that occur sequentially in a chain of speech or text. Given the same example sentence, there is a syntagmatic relationship between *The Sun* and *is shining*. Thus, syntagmatic relations reflect co-occurrences in a given context. In other words, syntagmatic relations concern positioning, while paradigmatic relations concern substitution. This aligns well with the notion that modelling taxonomic relations can help indicate which words can be replaced by other words, while taxonomic relations help in tasks such as predicting the next word in a sequence.

While this particular issue is beyond the scope of this thesis, it is worth noting that there is also discussion on whether both kinds of relations can be shared by the same pair of words. An article by Chiu and Lu (2015) analyses the relationship between paradigmatic and syntagmatic relations, with results suggesting that syntagmatic and paradigmatic relations between the same two words can coexist. Kacmajor and Kelleher make the same observation for taxonomic and thematic relations: although they are different and separate types of relatedness, the same pair of concepts can be connected by two different types of relatedness. An example of this is included in Figure 2.1, where *knife* and *fork* are taxonomically related

because they both belong to the category of *cutlery*, and are also thematically related because they perform complementary roles, for example in scenarios involving dinner[1].

In more practical terms, when it comes to modelling the two dimensions of semantic relationships, as a rule of thumb they are reflected in two different kinds of language resources: a **natural language corpus** primarily reflects thematic relationships between words by way of word co-occurrence, as they only provide linguistic context. Taxonomic relations, on the other hand, are rarely overtly expressed in examples of natural language. Though research has shown that such relationships can be automatically extracted from natural language corpora (Hearst, 1992), they are more accessible and more commonly modelled in the form of **knowledge-engineered language resources** such as thesauri, knowledge bases, ontologies, taxonomies and similar semantic networks, where relationships are reflected via explicit links between entities in the knowledge graph.

This distinction between taxonomic and thematic relatedness, as well as the different language resources they are reflected in, informs the theoretical basis of our work. It also informs some of the motivation behind our work, as we wish to explore the tension between different types of semantic information encodings by examining how the two different axes of semantic relations can be encoded in an embedding representation.

2.2 Embeddings

While general approaches to distributional semantics have historically been quite varied, the past decade has seen a convergence towards leveraging **vector space models**. First proposed by Salton et al. (1975), they truly began dominating the field of NLP around the early 2010's

[1] We acknowledge that this proposed mapping from taxonomic to paradigmatic and thematic to syngtagmatic is not perfect, and there is a more nuanced discussion to be had about the extent of the overlap in the terminology. However, we judge that the resemblance is sufficient for our purposes, as we simply use this as an analogy to further illustrate the concepts of taxonomic and thematic relatedness. Delving deeper into the terminological differences between these pairs of concepts falls beyond the scope of this thesis, and henceforth we shall exclusively rely on the terms *taxonomic* and *thematic*.

and have become the prevalent solution for representing the semantics of linguistic units. In a vector space model the meaning of a word is represented by a set of coordinates (i.e. a vector) that positions the word in a space, such that the relative location of the word with respect to other words reflects linguistic relationships between the words. In these models, words that have similar meaning have similar coordinates (i.e. vector representations). In essence, the contemporary approach to learning the appropriate coordinates for words in a vector space is to use neural network language models (NNLMs) trained on natural language corpora to produce the vector representations. Typically, NNLMs are constructed and trained as probabilistic classifiers with the goal of predicting probability distributions in a vocabulary. In other words, given some linguistic context, the neural network is trained to predict the probability of each word appearing in the sequence.

To make their probability predictions, such models use vector representations of words which they generate using standard neural network training algorithms such as stochastic gradient descent with back-propagation. These word representations are then obtained by first generating a vector representation with random values for each word, and then letting the algorithm update the values in the vectors during training with the goal of modelling the probability distribution of words in a corpus. This results in words that often occur together, or in similar contexts, having similar embeddings. However, instead of using NNLMs to produce actual probabilities, it is common to instead use the distributed vector representation encoded in the network's hidden layers as representations of words. Each word is then mapped onto its corresponding vector representation (Bengio, 2008).

In other words, based on the distributional hypothesis, the model maps words onto dense low-dimensional vectors by inferring the relative position of each word in a shared multidimensional semantic space from its context of use in the training corpus. The created continuous representations of words are then embedded in a shared vector space, hence why they are usually referred to as **embeddings**. The process of constructing embeddings

has undergone significant changes in the past decade with a myriad of new approaches continuously being developed. However, their fundamental property has remained unchanged: *distance in the vector space denotes a notion of (semantic) relatedness* (Schütze, 1993).

Generating embeddings results in a vector space that often contains meaningful substructures, as illustrated in Figure 2.2. For example, the vector representations for European capital cities can be found in a localised area of the space. Similarly, some models use the vector space to position the word vectors in such a way that meaningful relationships can be reflected via mathematical functions. Thus, they model semantic relations between words as linear combinations, capturing a form of compositionality that reflects the relational similarity between words. For example, some models allow for operations like the following: if the vector for *France* is subtracted from the vector for *Paris*, and then the vector for *Poland* is added, the resulting vector will be positioned nearby the vector for *Warsaw*. Similarly, if the vector for *car* is subtracted from the vector for *cars*, adding the vector for *apple* to the result will yield a vector that almost matches the vector for *apples* (Vylomova et al., 2016).

Most often, the sole source of embedding training data is a natural language corpus, meaning that embedding algorithms model their representations based on co-occurrence and positioning. It can thus be said that they are designed to model thematic relations. Indeed, many word embeddings have been shown to perform well on thematic similarity benchmarks (Baroni et al., 2014; Camacho-Collados and Pilehvar, 2018). On the other hand, taxonomic relations are not explicitly contained in natural language corpora and as such are not typically modelled by embedding algorithms, which show less success on stricter taxonomic and synonymic benchmarks (Hill et al., 2015; Kacmajor and Kelleher, 2019).

Evidently, learning word embeddings from only one of the two kinds of language resources provides an incomplete representation of the word as it models only one aspect of its meaning. Even though it is not difficult to argue that modelling both kinds of relationships is important for building AI with comprehensive natural language understanding abilities, most

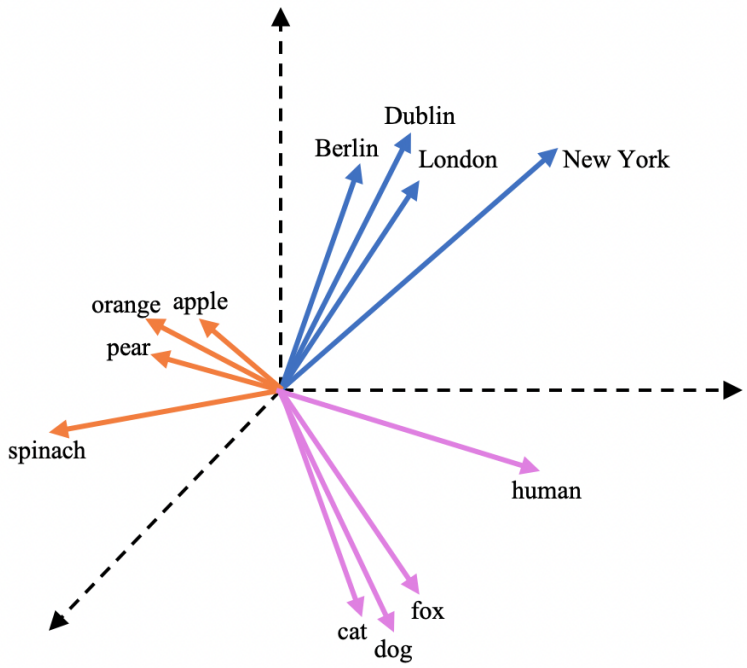


Figure 2.2 An illustrative example of how embedding models can, in principle, group semantically related words in close proximity in the vector space.

NLP models and systems, especially language embedding models, solely rely on natural corpora as their training resource. To remedy this, efforts have been made to transfer and integrate taxonomic information encoded in knowledge resources into distributed vector representations. We will elaborate on this in more detail in Chapter 4.

Finally, it is worth noting that within the vast amount of existing work related to vector space models there is also some variation in the terms that are used to refer to embeddings and related concepts. While a given author's choice might sometimes depend on the work's perspective and focus of interest, in the majority of cases the different terms are synonymous, or near-enough to make little difference, as ultimately the referent is always the same—a multidimensional real-valued vector generated by some kind of statistical or machine learning model. Hence in this thesis we will be using a number of terms interchangeably to refer to embeddings and the models that generate them. While we will most often use the term embeddings, on occasion we might resort to terms such as *encodings*, *dense low-dimensional*

vectors, *dense embeddings*, *distributed representations*, *distributed vector representations*, *distributed word representations* or *distributed meaning representations*. Similarly, we will most often refer to the models that generate them as embedding models, however sometimes we might also use the terms *embedding algorithms* or *encoders*.

It is also important to highlight some terminological nuances which might otherwise be taken for granted. The concept of *dense vectors* contrasts with the concept of *sparse vectors*. Sparse vector representations derive their name from the fact that they are sparsely populated with information. Typically they would have a high number of dimensions where most of the dimension values would be set to zero, with only a handful containing informative values. An example of this is a one-hot encoding vector, which can be used to represent a sentence: given a vocabulary of words, it encodes words that appear in the sentence with 1 and words that do not with 0. Thus the number of dimensions in the vector are equal to the size of the vocabulary, which is typically in the range of tens of thousands, while the number of informative vector dimensions matches the size of the sentence. In contrast, NNLMs generate dense vectors, where there are far fewer dimensions (e.g. only 300 or 768) and each dimension holds relevant information, and in principle no dimension value is ever set to 0. Furthermore, a common property of dense vectors generated by NNLMs is that the semantics assigned to each of their dimensions is opaque; in fact, the encoding of a single concept is often distributed across multiple dimensions, and a single dimension is capable of representing more than one concept. This *distributed* [2] property of the vector representation is in contrast with a *localist* representation: in a sparse one-hot-encoding vector, there is a one-to-one correspondence between concepts and dimensions, as each dimension encodes a single piece of information, e.g. the presence or absence of one word from the vocabulary

[2]Not to be confused with distributional representations, which can be considered a subset of distributed representations (Ferrone and Zanzotto, 2020), as they only refer to language vectors that are based on the distributional hypothesis, describing information related to the contexts in which they appear. Whereas distributed representations can be used to encode extra-linguistic information and thus have no relation to the distributional hypothesis of language, but are a more general type of vector representation.

(Kelleher, 2019, page 129). With that in mind, in this thesis we will be working exclusively with **dense distributed vectors**.

Having established the core terminology and theory behind it, we now take the opportunity to introduce three influential thematic embedding models that will be used throughout the thesis.

2.2.1 word2vec

One could argue that the publication of word2vec (Mikolov et al., 2013a,b) has most strongly impacted the landscape of distributional semantics in NLP. As one of the earlier examples of a distributed word representation model that learns representations using a neural network, it became widely popular upon its release and has shaped the trajectory of the field, inviting comparisons to this day, even while far superior models have been developed since. Word2vec is based on a feedforward neural architecture which is trained with a language modelling objective. Mikolov et al. proposed two different but related word2vec models: CBOW (Continuous Bag of Words) and SGNS (Skip-Gram with Negative Sampling). We provide an illustration of the different word2vec architectures, as shown in Figure 2.3, given the example sentence *The chef prepared the meal*.

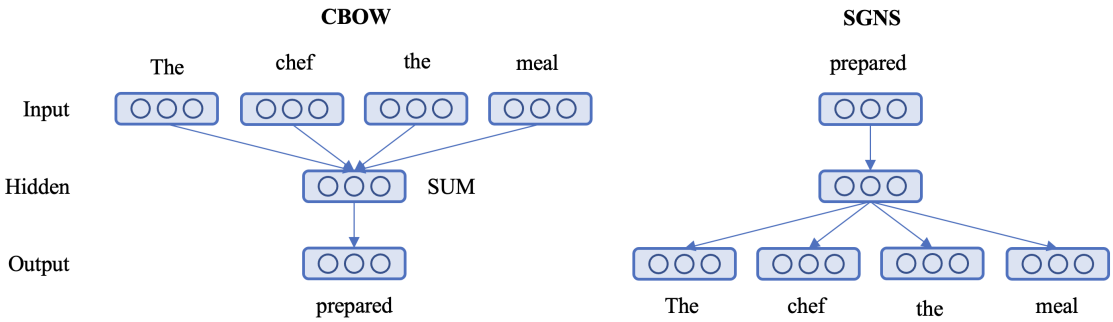


Figure 2.3 Learning architecture of the CBOW and Skipgram models of word2vec. The illustration is based on Figure 1 in (Mikolov et al., 2013a).

CBOW is designed so that it would predict a target word using the input of its context words within a sliding window of n words. So in our example sentence, to predict the

word *prepared*, it uses its immediate context words *The*, *chef*, *the* and *meal*. Architecturally, CBOW is similar to a feedforward NNLM, where the non-linear hidden layer is removed and the projection layer is shared for all words in the context window, thus all words are projected into the same position (their vectors are averaged). The objective function is a log-linear classifier which predicts the middle word given the past $n/2$ history words and $n/2$ future words at input (empirically, it seems the best results are obtained by using $n = 8$). It is called a bag-of-words model as the order of words in the history does not influence the projection—there is no relevance of the position of the word in determining the vector of the middle word.

In a sense, the SGNS model is an inverted version of CBOW (as illustrated in Figure 2.3), where instead of predicting the middle word based on the context, it tries to predict the word's context words using the target word as input. So in our example sentence, the model's input would be the vector for the word *prepared*, and its goal would be to predict the context words *The*, *chef*, *the* and *meal*. Note that SGNS assumes that a focus word occurring in a text depends on the words the focus word co-occurs with inside a fixed-sized context window, but that those context words occur independently of each other. This conditional independence assumption in the context words makes computation more efficient and produces vectors that work well in practice. SGNS uses the current word as an input to a log-linear classifier with a continuous projection layer, which predicts words within the window before and after the current word.

The negative sampling aspect of the SGNS algorithm is a way of producing "negative" context words for the focus word by simply drawing random words from the corpus. These random words are assumed to be incorrect context words for the focus word. The positive and negative examples are used by an objective function that seeks to maximise the probability that the positive examples came from the corpus whilst the negative examples did not. In our experiments we will use the SGNS word2vec architecture, as it has been shown to outperform

CBOW on a number of relevant tasks and seems to be more consistently used in the literature, likely due to the benefits of using negative sampling.

2.2.2 GloVe

Another prominent word embedding architecture we will employ in our work is GloVe (Pennington et al., 2014), which stands for Global Vectors. GloVe is an unsupervised log-bilinear regression model trained to learn word representations on aggregated global word-word co-occurrence statistics from a natural language corpus, which yields a vector space with meaningful sub-structures. GloVe differs from both word2vec architectures in that, instead of predicting a target word or its context, it is designed to predict a given word's global co-occurrence statistics from the training corpus. The architecture essentially combines features of global matrix factorisation and local context window methods. Pennington et al. claim that both families suffer significant drawbacks individually and point out that methods like SGNS poorly utilise corpus statistics on a global level, which is the type of information that GloVe is designed to leverage.

The main idea behind GloVe is that the ratio of co-occurrence probabilities of two words, w_i and w_j, with a third probe word w_k, i.e., $P(w_i, w_k)/P(w_j, w_k)$, is more indicative of their semantic association than a direct co-occurrence probability, i.e. $P(w_j, w_j)$. Using these global co-occurrence statistics, they propose an optimisation problem which aims at fulfilling the following objective:

$$w_i^T w_k + b_i + b_k = log(X_i^k) \tag{2.1}$$

where b_i and b_k are bias terms for word w_i and probe word w_k, and X_i^k is the number of times w_i co-occurs with w_k. Fulfilling this objective minimises the difference between the dot product of w_i and w_k and the logarithm of their number of co-occurrences. This optimisation

results in the construction of vectors w_i and w_k whose dot product provides a good estimate of their transformed co-occurrence counts.

Pilehvar and Camacho-Collados (2020) highlight that, while GloVe does not make use of neural networks, it is still considered to be a predictive model, rather than a count-based model. Its architecture is different from conventional count-based models in that it starts with a randomly initialised vector and uses stochastic gradient descent to update the vector based on the error in predicting co-occurrence, optimising a non-convex objective so that words that co-occur often end up with similar vectors. In this sense, GloVe also significantly diverges from word2vec, and their difference is additionally compounded by the fact that, while GloVe still uses context windows, it does so globally, rather than individually, and does not rely just on local statistics, i.e. local word context information, like word2vec does, but also incorporates global statistics, i.e. word co-occurrence statistics across all words in the corpus.

An important aspect of pretrained word embedding models such as word2vec and GloVe is that they provide a single, **static embedding** for each word in a vocabulary. The word representation is then fixed and is essentially independent from the context in which the word appears, thus conflating all possible alternate meanings into one representation. This has always been one of the biggest criticisms of these approaches, as they completely disregard phenomena like homonymy and polysemy, where the same surface form can take on multiple, sometimes completely disparate meanings depending on the context. In addition to ignoring the role of a word's context in shaping its meaning, restricting the representations to individual words makes it difficult to represent higher order semantic phenomena such as compositionality and long-distance dependencies.

2.2.3 BERT

Fortunately, it only took about half a decade since embeddings gained universal prominence for these issues to be resolved, ushering in another significant paradigm shift in NLP: **contextual embeddings**. In contrast to static word embeddings, contextual word embeddings are dynamic in the sense that the same word can be assigned different embeddings if it appears in different contexts. This is possible because contextual embeddings are assigned to tokens as opposed to types. Instead of receiving words as distinct units and providing independent word embeddings for each, contextual models receive the whole text span (the target word along with its context) and provide specialised embeddings for individual words which are adjusted to their context. While there had been earlier attempts at addressing the issue of meaning conflation via building contextual embeddings (Li and McCallum, 2005; Melamud et al., 2016), including a number of prominent LSTM-based architectures (Peters et al., 2017; McCann et al., 2017; Peters et al., 2018b), the true turning point came with the advent of the novel **transformer** architecture (Vaswani et al., 2017).

The transformer model is an auto-regressive sequence transducer: its goal is to convert an input sequence to an output sequence, while the predictions are done one part at a time, consuming the previously generated parts as additional input. Similarly to most other sequence-to-sequence models, the transformer employs an encoder-decoder structure. However, unlike previous models, the transformer forgoes the recurrence of recurrent neural networks (RNNs) for a fully feedforward attention-based architecture. Self-attention is a special attention mechanism which looks for relations between positions in the same sequence. Its goal is to allow the model to consider the context while "reading" a word.

According to Pilehvar and Camacho-Collados (2020), transformers come with multiple advantages over RNNs, which were previously the dominant models: (1) compared to RNNs, which process the input sequentially, transformers are parallel which makes them suitable for GPUs and TPUs which excel at massive parallel computation; (2) unlike RNNs, which

have memory limitations and tend to process the input unidirectionally, thanks to the self-attention mechanism transformers can attend to contexts relating to a word from distant parts of a sentence, both before and after the words appearance, in order to enable a better understanding of the target word without any locality bias.

This new model architecture showed much promise and was soon applied to many ML domains, including sequence encodings, and a number of modified versions of the transformer have been developed since, specifically applied to language modelling. Alongside OpenAI's GPT model (Radford et al., 2018), which has the limitation of only attending to previously seen tokens in the self-attention layers, arguably the most prominent transformer-based language model is BERT (Devlin et al., 2019), which almost instantly spawned the rapidly growing field with a tongue-in-cheek name, BERTology (Rogers et al., 2020). BERT's essential improvement over GPT is that it provides a solution for making transformers bidirectional. This addition enables it to perform a joint conditioning on both left and right context in all layers. This is achieved by changing the conventional next-word prediction objective of language models to a modified version, called masked language modelling, where instead of predicting the next token, the model is expected to guess a token that is randomly masked from the input seqence, using information from the unmasked remainder of the sentence. This allows the model to have conditioning not only on the right (next token prediction) or left side (previous token prediction), but on context from both sides of the token being predicted.

There is an additional aspect to BERT that further distinguishes it from conventional static word embedding models such as word2vec and GloVe: while these static embeddings take whole words as individual tokens and generate an embedding for each token, usually resulting in hundreds of thousands or millions of token embeddings, BERT segments words into subword tokens and generates embeddings for these subword units instead. Segmenting words into subword units offers a number of advantages: (1) it drastically reduces the

vocabulary size, from millions of tokens to dozens of thousands; (2) it provides a solution for handling out-of-vocabulary words as any unseen word can theoretically be re-constructed based on its subwords (for which embeddings are available); (3) it allows the model to share knowledge among words that have similar surface structures, with the assumption that they have a shared semantics (Pilehvar and Camacho-Collados, 2020).

In our work we use BERT as a representative of contextual language models. This allows us to consider three different types of predictive embedding models: **word2vec** as an example of the standard NNLM architecture, **GloVe** as a log-bilinear regression model and **BERT** as an example of a transformer-based contextual model. Studying three different types of embeddings will make comparisons more valuable, as their differences can inform our result interpretation. Notably, one characteristic that the models do share is that their resulting embeddings are not human-interpretable. This is symptomatic of all deep learning models, which are widely known to be black boxes (Alishahi et al., 2019), as it is difficult to investigate the "reasoning" behind their decisions. This is why, in parallel to the staggering developments in machine learning models, the field of model interpretability has developed alongside it, working towards explaining the decisions the models make. In the case of distributional semantics, the pertinent topic is interpreting vector representations and the types of information they might be capturing. This forms the third aspect of our work, providing us with a methodological framework, as well as informing the questions that shape our hypotheses.

2.3 Probing

With the aim of interpreting embedding models and distributed meaning representations, the notion of **probing** has gained considerable traction in the NLP community. Intriguingly, it seems the framework has been concurrently, yet independently proposed by different groups of researchers: Ettinger et al. (2016) presented proof-of-concept preliminary experiments

that propose a *diagnostic method* for probing specific information captured in vector representations at the sentence level. They describe their method as "linguistically-motivated and computationally straightforward", directly testing for extractability of semantic information that is being captured in sentence representations by using them as training data for a classifier. Similarly, Veldhoen et al. (2016) developed a tool called *diagnostic classifiers*, the goal of which is to read out whether certain information is present in the hidden representations of a neural network and make a prediction about the hierarchical semantics in the sentence being represented. Finally, Adi et al. (2017) introduced what they call *auxiliary prediction tasks*, a framework that can facilitate a better understanding of encoded sentence representations. By defining prediction tasks around isolated aspects of sentence structure (such as length, word content, and word order) they score representations by the ability of a classifier to solve a given task when using the representation as input.

Functionally, the proposed approaches are almost identical, with only minor implementation and application differences. Generally, the common thread between them can be described as: training a classifier over embeddings produced by a pretrained model, and assessing the embedding model's knowledge encoding via the probe's performance. Given this framing, it is worth noting that around the same time similar diagnostic work was being carried out by a number of other researchers, though they did not explicitly name their framework. For instance, Köhn (2016) used the performance of a simple linear classifier trained on embeddings as a proxy for how well those embeddings will perform when used in a syntactic parsing task. Similarly, Shi et al. (2016) tested whether different sequence-to-sequence machine translation systems learn to encode syntactic information about the source sentence in English, by using the model's hidden states to predict syntactic labels of source sentences via a logistic regression classifier. Finally, while Salton et al. (2016) did not overtly apply the diagnostic framework to their work, they employed the same pipeline: they used a classifier

trained on sentence embeddings to predict figurative usage in the sentence, making inferences about what kind of information is encoded within the representations.

However, a crucial difference between the early work that overtly names the framework and work that does not, is in the intention—Köhn (2016) wished to simplify a computationally expensive syntactic parsing task, while Salton et al. (2016) aimed to build an idiom token identification model, and the revelation about linguistic information being encoded in the embeddings was incidental. Another difference is that the early probing work emphasises that a meticulous construction and curation of the probing task dataset is necessary to facilitate an unambiguous interpretation of what might be encoded in the embeddings. Ettinger et al. (2016) describe their sentence datasets as "controlled and annotated as precisely as possible for their linguistic characteristics". The same sentiment is echoed by Conneau et al. (2018), who posit that a probing task should ask a simple, unambiguous question in order to minimise interpretability problems. If constructed with the goal of simplicity, it is easier to control for biases in probing tasks than it is in downstream tasks.

The work by Conneau et al. (2018) gained a lot of traction as they applied the probing framework to a large number of models, as well as developed and released a large set of diverse probing tasks, making it more accessible for researchers to enter this research space. Arguably, it was the popularity of their work that made probing language representations a commonplace interpretability technique in NLP, as it has since gained significant momentum and has been used to explore many different aspects of text encodings (e.g. Hupkes et al. (2018); Giulianelli et al. (2018); Krasnowska-Kieraś and Wróblewska (2019); Tenney et al. (2019a); Lin et al. (2019); Şahin et al. (2020); Voita and Titov (2020); Garcia et al. (2021)).

A significant contributor to its popularity is also the inherent modularity of the probing pipeline: it is agnostic with respect to the encoder architecture, or indeed any other one of its required elements. This makes it attractive and easy to work with, as it can be applied to a large number of varying scenarios—it is a simple matter of plugging various components into

the pipeline, be it different embedding models, different probes or different linguistic tasks. In theory, the probing framework can be used to assess any property of language contained in a linguistic unit representation (word, phrase, sentence) that can be expected to be encoded by an embedding model. And indeed, much related work has been done studying the types of linguistic information that can be encoded in language representations: probes trained on various embeddings have been used to successfully predict surface properties of sentences (Adi et al., 2017; Conneau et al., 2018), part of speech and morphological information (Belinkov et al., 2017a; Liu et al., 2019a), as well as syntactic (Zhang and Bowman, 2018; Peters et al., 2018a; Liu et al., 2019a; Tenney et al., 2019b), semantic (Belinkov et al., 2017b; Ahmad et al., 2018), and even number (Wallace et al., 2019), discourse structure (Chen et al., 2019) and world knowledge information (Ettinger, 2020), among others (Belinkov and Glass, 2019).

Seemingly more studies have been devoted to probing for syntactic than semantic phenomena, especially in BERT, which is often the prime suspect in the majority of recent studies (Rogers et al., 2020). This lack of focus on semantics is likely due to the fact that it is difficult to narrowly define a simple, unambiguous semantic probing task and curate a dataset that would facilitate a straightforward interpretation within the probing framework. Granted, while underrepresented, some semantic probing work has been done: BERT has been shown to encode information about entity types, relations and semantic roles (Tenney et al., 2019b), and has demonstrated the ability to prefer the incorrect fillers for semantic roles that are semantically related to the correct ones, over those that are unrelated; for example *to tip a chef* is preferred over *to tip a robin*, but not as desirable as *to tip a waiter* (Ettinger, 2020). Additionally, the survey by Belinkov and Glass (2019) does highlight a number of studies which, while they do not all use the probing framework explicitly, are semantics-related. This includes work on emotions (Qian et al., 2016), lexical semantics (Belinkov et al., 2017b), and word sense disambiguation (Ahmad et al., 2018). The aforementioned work by Salton

et al. (2016) also operated in the domain of semantics, showing that Skip-Thought vectors can be used to predict idiomatic usage, indicating they encode some representation of idiomaticity. Nedumpozhimana and Kelleher (2021) use the probing framework to show that BERT's idiomatic key is primarily found within an idiomatic expression, but also draws on information from the surrounding context. However, conversely, Garcia et al. (2021) claim in their work that contextual embeddings like BERT and ELMo do not yet accurately represent idiomaticity. We discuss this further in Chapters 6 and 8, where we highlight our own contribution to the space of probing for semantic information and figurative language.

2.3.1 Categories of Probing Work

With the rapidly growing body of work in the field of probing, a number of dichotomies have begun to emerge relating to the ways that probing research is being done. Though all the work inherently belongs to the area of interpretability, with the general aim of better understanding embedding models, there are also some nuances in the starting positions of research that rely on slightly different presuppositions.

For instance, Ravichander et al. (2020) distinguish different points of view on what embeddings are, highlighting a difference between work that is *instrumentative* and work that is *agentive*. The instrumentative perspective treats embedding models as tools used to mine or store linguistic knowledge from text. When viewed as such, the primary purpose of probing work is to identify effective techniques to extract information from these embeddings so that they can be used in downstream NLP pipelines. In contrast, a significant amount of research adopts an agentive perspective, where embedding models are treated as AI agents that have certain linguistic competencies and world knowledge that can be analysed through tasks such as natural language inference or story completion.

Meanwhile, Vig et al. (2020) consider probing to be a method of analysis and distinguish two different types of analysis methods: *structural* and *behavioural*. Structural analyses

aim to shed light on the internal structure of a neural model through probing classifiers that predict linguistic properties using representations from trained models. Behavioural analyses, on the other hand, aim to assess a model's behaviour via its performance on constructed examples in tasks such as natural language inference. Note the similarities between structural and instrumentative analysis, as well as behavioural and agentive—while they do not map perfectly to one another, the two distinctions are analogous to a degree. For example, both the behavioural and agentive perspectives seem to rely on a more cognitive-science-based approach to studying the behaviours of NNLM embedding models, treating them as agents whose performance can be examined on experimental tasks borrowed from psycholinguistics and cognitive science literature. The same tasks are also used for experiments involving human participants, thus agentive and behavioural probing work, explicitly or implicitly, draws comparisons with human language competencies and performance.

Finally, we have identified another such dichotomy in the literature, though it is more implicit. The perspective relates to the nature of the linguistic information encoded in a representation: there is a tension between identifying the mere *presence* of information, versus its *extractability*. This tension exists because the probing framework relies on two separate sets of models: the encoder which creates the language representation, and the probe which is expected to be able to read this information and use it to make predictions. However, even if an encoder does read some linguistic information from its input and stores it in its embedding, it does not necessarily follow that this information will be easily recoverable by another system. In addition, the recoverability of the information depends on the quality of the probe, but also on the way the encoder has structured the information in the representations.

This issue of *ease of extraction* has been discussed by Pimentel et al. (2020) and Voita and Titov (2020), who propose different flavours of information-theoretic approaches to probing that measure the "amount of effort" needed to train a successful probe. They argue that "better" representations should make their respective probes easily learnable, and

consequently make their encoded information more accessible. So if the research goal is to show the presence of some information in a representation, then the results will allow for inferences on the embedding model, rather than the probe. On the other hand, if the goal is to leverage this information in downstream tasks, then the relevant property of the information itself is its ease of extraction, rather than its presence, and inferences will be more dependent on the probing classifier that is used. Also note that this tension could be considered a subset of the structural approach, as it considers the question of whether information is structured in such a way that it can be easily leveraged by downstream users of embeddings. In that sense, it is also instrumentative, as it considers embeddings to be tools that store information in a certain way.

In this chapter we have cited a mix of work representing each of these types of approaches, however we do not attempt to categorise all related work nor do we go into much more detail on this, but rather simply highlight that these differences exist and take the opportunity to explicitly state which of the possible perspectives we take in our work. Indeed, while there are a number of available perspectives and avenues to pursue, in our case we position our work within the existing literature as being **instrumentative**, i.e. we view embeddings as tools that extract and store knowledge from text; we consider our probing method to be **structural**, i.e. it provides insight into how information is encoded within the representation and the vector space; and the goal of our work is to identify the **presence** of information in embedding components, rather than its extractability. Signposting this also allows us to better define the scope of the thesis: for example, while we acknowledge the importance of the questions raised by Pimentel et al. (2020) and Voita and Titov (2020) and their findings, we do not adopt their information-theoretic perspective as their work is more concerned with extractability, whereas we attempt to identify presence, for which our framework is sufficient.

2.3.2 Limitations of Current Probing Methods

While work in area of probing flourishes, it does not come without its limitations. We highlight that it is easy to misconstrue the basic inference that the framework rests on, which often seems to be: "if the probe performs well on the probing task, this indicates that the relevant knowledge is encoded in the probed representation". If this were the case, it would raise several questions, not least of which is: "How does one determine if the probe performs well?" It is not feasible to expect that any given probe's evaluation score will provide valuable insights about the embedding based solely on the one accuracy value, as it is not possible to decouple the contribution of the representation and the contribution of the classifier. Thus the prototypical probing pipeline is insufficient to provide any tangible insight into one embedding model alone and, at the very least, two different representations are needed to make viable inferences. Though this does not seem to be explicitly stated anywhere, probe interpretations such as made by Conneau et al. (2018) are in fact relative and dependent on differences between representations, so the basic inference would have to be "if the same probe predicts the task better using representation A than using representation B, this indicates that representation A is better at encoding the relevant knowledge."

While probe interpretations rely on differences in representations, there is evidence that some probes fail to adequately reflect such differences: Zhang and Bowman (2018) used probes to compare pretrained representations with randomly initialised ones and in some cases had to reduce the amount of probe training data in order to observe differences in the probe's accuracy with respect to the random baselines. A related result was found by Hewitt and Liang (2019), who include a "control task" setting in the probing pipeline by probing for labels randomly associated to word types, which has shown that under certain conditions, above-random probing accuracy can be achieved even when the information that one probes for is linguistically-meaningless noise. Additionally, Ravichander et al. (2021) show that text encoders can learn to encode linguistic properties even if they are not needed for the task on

which the model was trained. Through a set of controlled synthetic tasks, they demonstrate that embedding models can encode these properties considerably above chance-level even when distributed in the data as random noise, further challenging the common interpretation of probing.

Furthermore, given this reliance on multiple encoders, most probing evaluations are, in a sense, extrinsic, as most related work compares a number of different embedding models and then draws conclusions based on their differences. Work concurrent with ours (Torroba Hennigen et al., 2020) highlights a distinct lack of an intrinsic probe evaluation setting, as result interpretations are always relational and dependant on differences between different encoders. Certainly, if the goal is to compare different encoders, then the probe's performance can inform which model is better than others at storing the information. However, if the goal is to examine whether a particular representation encodes some information at all, to perform an intrinsic evaluation, then the single evaluation provided by the probe is not sufficient to give a reliable answer.

Recent work addresses some of the above problems. Torroba Hennigen et al. (2020) develop an intrinsic probe that is focused on isolating the dimensions that encode relevant information in the embedding vectors. Furthermore, Feder et al. (2020) construct counterfactual representations in order to compare the performance of the probe with and without the pertinent information. Similarly, Elazar et al. (2020) use Iterative Null-space Projection (Ravfogel et al., 2020) to remove the relevant information from the representation, allowing a comparison of probe performance with and without the removed information, thus allowing to measure the effects of confounding factors. In essence, these recent efforts address the issue of relativising probe interpretations by removing information from the encoding. Note that in this case, rather than referring to interpretations of differences between other model architectures, the term *relative interpretation* refers to an intrinsic evaluation, comparing the model to altered versions of itself.

In that sense, this thesis finds its place alongside this body of work, as our method allows for an intrinsic evaluation of a single embedding model while still allowing for a relativised probe interpretation. We describe the method in detail in the following chapter, where we will also highlight differences between our work and a number of related methods.

Chapter 3

Method: Probing With Noise

Probing in NLP, as defined by Conneau et al. (2018), is a classification problem that predicts linguistic properties using dense embeddings as training data. The framework rests on the assumption that the probe's (relative) success at a given task indicates that the encoder is storing readable information on the pertinent linguistic properties. With the ability to provide such insight into embeddings, probing has quickly become an essential tool for encoder interpretability.

The typical probing pipeline is as follows:

1. Choose a probing task (e.g. predicting the voice of the main verb in a sentence)
2. Choose or design an appropriate dataset (e.g. a set of sentences with active/passive labels)
3. Choose a word/sentence representation (i.e. the embedding)
4. Choose a probing classifier (the probe)
5. Train the probe on the embeddings as input
6. Evaluate the probe's performance on the task

The final step is an evaluation of the probe's performance, based on which inferences can be made regarding the presence of the probed information in the embeddings. The main inference is that if the probe performs well on the probing task, this is an indication that the relevant knowledge might be encoded in the probed representation. This way, different

encoders can be compared and the probe's relative performance can inform which model stores the information more saliently.

However, if the goal is to examine properties of a particular representation, to perform a kind of intrinsic evaluation, then the probe's accuracy alone cannot provide such insight. At best, it indicates that the encoding might contain non-zero amounts of information on the relevant property, but still does not distinguish between what comes from the probe and what comes from the embeddings. Yet there is a range of possible insights that can be gleaned by delving a little deeper. With the goal to better understand embeddings and how they encode information, we investigate their geometric properties, with a focus on the role of the norm, and ask the questions: Where in an embedding can information be contained? In what way are linguistic properties encoded in vector space? Are different properties encoded differently? Do different encoders store information in different places?

To address these questions, in this chapter we first analyse the geometric structure of embeddings and identify components which can encode information, which we call *information containers*. We then use this understanding to extend the existing probing framework so that it can identify which information container encodes the pertinent information. We do this by employing an ablation method using targeted noise injections into the embeddings that disrupt each information container. Finally, we walk through a hypothetical application of the method to give an example of what kinds of results and insights this method can provide, before applying it to real datasets in Chapters 5, 6 and 7.

3.1 Information Containers

In essence, embeddings are just vectors positioned in a shared multidimensional vector space. Vectors, as opposed to scalars, are geometrically defined by two aspects: having both a **direction** and **magnitude** (Hefferon, 2018, page 36). Direction is the position in the space that the vector points towards (expressed by its dimension values), while magnitude is a

vector's length, defined as its distance from the origin of the space (expressed by the vector norm) (Anton and Rorres, 2013, page 131). Individual dimensions can be considered local properties of vectors, while the norm can be considered a distributed property, a function of the full set of dimensions.

Modelling linguistic items by assigning them vectors allows us to use a geometric metaphor for meaning, as "vector similarity is the only information present in Word Space: semantically related words are close, unrelated words are distant" (Schütze, 1993, page 896). While more recent work has found that it is possible for additional meaningful substructures to be found in vector space (Hernandez and Andreas, 2021), Schütze's statement still forms the foundation of our understanding that distance in a geometric vector space model can be interpreted as analogous to (semantic) relatedness.

Most commonly, the cosine similarity measure is used as a proxy for similarity between two vectors (Widdows and Cohen, 2015). It normalises vector length and compares the cosine of the angle between two given vectors to determine whether they are pointing in roughly the same direction. If two vectors have a high cosine similarity, this is interpreted as the units they represent being similar as well.

However, Karlgren and Kanerva (2021) point out that when calculating cosine similarity, due to the prerequisite step of vector normalisation, the points of interest in the high-dimensional space are scaled to fall on the surface of a hypersphere. This means that a search for structure (i.e. similarity) in high-dimensional space is actually a search for structure as it is projected on to the surface of a hypersphere. The issue is that, when the space has high dimensionality, an increasing majority of the points lie far from the surface of the hypersphere. Consequently, any structure in the original space that depends on the differences in distance from the origin is lost in the projection. In other words, if the vector norm were to carry any meaning itself, then calculating a cosine similarity measure does not account for this information at all. In fact, Goldberg (2017, page 117) mentions that for many word

embeddings normalising the vectors removes word frequency information, noting that "this could either be a desirable unification, or an unfortunate information loss".

One could easily argue that this is not a major concern: it is well understood that information contained in a vector representation is encoded in its dimension values, primarily the vector direction, rather than magnitude. This is corroborated by the majority of interpretability research focused specifically on dimensions, where a number of probing studies research their role as carriers of specific types of information (e.g. Karpathy et al. (2015); Qian et al. (2016); Bau et al. (2019); Dalvi et al. (2019); Lakretz et al. (2019)). Work by Torroba Hennigen et al. (2020) shows that most linguistic properties are reliably encoded by only a handful of dimensions. This finding is consistent with the results of Durrani et al. (2020), who analysed individual neurons in pre-trained language models and also found that small subsets of neurons are sufficient to predict certain linguistic tasks, with lower-level tasks (i.e. morphology) localized in fewer neurons, compared to a higher-level syntactic task.

However, information can be encoded in a representational vector space in more implicit ways, and relations can be inferred from more than just vector dimension values. Embedding vectors have varying magnitudes and can be scattered around the vector space at different distances from the origin. While the norm is a distributed property of a vector's dimensions, it not only relates the distance of a vector from the origin, but indirectly also its distance from other vectors. In fact, two vectors could be pointing in the exact same direction, but their distance from the origin might differ dramatically[1]. We suspect that in semantic vector spaces, vectors which are closer to the origin might have properties in common compared to vectors that are far away from it. Hence, analogously to the angle between vectors reflecting their relationships, it should be possible for the vector magnitude—or norm—to act as an implicit container of information as well.

[1]Mathematically, two vectors can only be considered equal if both their direction and magnitude are equal (Anton and Rorres, 2013, page 137).

While it is underrepresented relative to research on vector dimensions, the effect of the norm encoding certain types of information has occasionally been observed in the literature: as noted earlier, according to Goldberg (2017), for many word embedding algorithms the norm of the word vector correlates with the word's frequency. For example, in fastText embeddings (Mikolov et al., 2018) the vectors of stop words (the most frequent words in English) are positioned closer to the origin than content words (Balodis and Deksne, 2018). Though this was not the focus of their work, Adi et al. (2017) briefly examined the relationship between sentence length and norm and have found that in sentence representations derived from averaged word2vec word vectors the embedding norm decreases as sentences grow longer. Additionally, Hewitt and Manning (2019) investigated structural properties of the word representation space, with a sole focus on how syntactic information is encoded in vector space, and found that the structure of syntax trees emerges through properly defined distances and norms in BERT and ELMo's word representation spaces. Recent research (Kobayashi et al., 2020) also highlights the relevance of the norm during the embedding training process, demonstrating that it plays an integral part in BERT's attention layer, controlling the levels of contribution from frequent, less informative words, such as stop words, by controlling the norms of their vectors.

Taken together, these findings seem to indicate that vector magnitude is a vector property which could be leveraged by embedding models to encode linguistic information. However, it seems that these results have not been followed to their logical conclusion, which we argue here explicitly as: a vector representation has two separate **information containers**—vector *dimensions* and the vector *norm*.

This prompts the question: if the norm can encode word frequency, sentence length, and syntactic tree structure, which other linguistic properties of words or sentences can be stored there? In this thesis we test the hypothesis that the two containers can be used to encode

different types of information, and offer a systematic and comprehensive exploration of the types of information a vector norm can encode across different encoders.

To study this, we require a probing method that provides an intrinsic evaluation of any given embedding representation, for which the typical probing pipeline (as described above) is not suited. We thus extend the existing probing framework by introducing random noise into the embeddings. This enables us to intrinsically evaluate a single encoder by testing whether the noise disrupted the information in the embedding being tested. The right application of noise enables us to determine which embedding component the relevant information is encoded in, by ablating that component's information. In turn, this can inform our understanding of how certain linguistic properties are encoded in vector space, providing novel geometric insights into embeddings[2].

3.2 Probing with Noise

Our addition to the probing pipeline is incorporated as steps 7 and 8:

1. Choose a probing task
2. Choose or design an appropriate dataset
3. Choose a word/sentence representation
4. Choose a probing classifier (the probe)
5. Train the probe on the embeddings as input
6. Evaluate the probe's performance on the task **(vanilla baseline)**
7. **Introduce systematic noise in the embedding**
8. **Repeat training, evaluate and compare**

Though this may seem like a minor addition, it changes the approach conceptually. Now, rather than providing the final answer, the output of step 6 establishes an intrinsic, *vanilla baseline*. After each iteration of noise, the embeddings with noise injections can be compared

[2]It is important to note that even if there is a distinction between information encoded in the dimension and norm containers, in order to successfully probe for it, this information needs to be accessible to the probing classifier when doing a probing task. This requires the probe in question to be able to take a global view of the input features: e.g. decision trees test elements one at a time and so would not have access to the norm, but a fully-connected MLP would.

against the vanilla embeddings in steps 7 and 8, which offers a relative intrinsic interpretation of the evaluation. In other words, using relative information between a vector representation and targeted ablations of itself allows for inferences to be made on where information is encoded in embeddings.

The method relies on three supporting pillars: (a) random baselines, which in tandem with the vanilla baseline provide the basis for a relative evaluation; (b) statistical significance derived from confidence intervals, which informs the inferences we make based on the relative evaluation; and (c) targeted noise, which enables us to examine where the information is encoded. We describe them in the following sections, starting with the noise.

3.3 Choosing The Noise

The nature of the noise is crucial for our method, as the goal is to systematically disrupt the information containers in order to identify which information the disrupted container is related to. We use an ablation method to do this: by introducing noise into either container we "sabotage" the representation, in turn identifying whether the information we are probing for has been removed. It is important that the noising function applied to one container leaves the information in the remaining component intact, otherwise the results will not offer insight into which container the information is in.

3.3.1 Ablating the Dimension Container

The noise function for ablating the dimension container needs to a) remove its information completely, while not modifying the norm in any way; and b) should also not change the dimensionality of the vector, in order to control for the confounding factor of overfitting—it is possible that as the dimensionality of a feature space increases, the chance of the probe finding a random or spurious hyper-plane that performs well on the data sample also increases

(Hewitt and Liang, 2019). Maintaining the dimensionality ensures that the probability of the model finding such a lucky split in the feature space remains unchanged.

There are a number of ways to directly intervene in a vector's dimensions: we could simply delete a number of dimensions and their values from the vector. However, this reduces the dimensionality of the vector space and changes the norm of the vectors, making comparisons to an unmodified baseline embedding problematic. We could retain the dimensionality of the vector space while still removing information from specific embeddings by changing the dimension values to zero, rather than removing them altogether. However, any change in values also modifies the vector's norm, so such a modification is not an appropriate candidate to probe this information container.

One option that circumvents this conundrum is to apply a transformation to the vector by shuffling the values in any given vector, randomly reassigning them to different dimensions. Applying a different random shuffle to each vector would dissociate any individual vector dimension from any particular type of information. This would invalidate any semantics assigned to a particular dimension, as dimension values become inconsistent across different vectors, while the actual values, as well as the norm as a distributed property, remain unchanged.

In principle, we expect this approach would suffice to fully and exclusively remove dimension information. However, the approach does not generalise well, nor is it particularly rigorous: the actual dimension values are still present in the vector and, while it is unlikely, it is still possible that given a powerful enough probe and a high enough number of samples and epochs, a signal might still be extractable from the randomised values.

Instead, we apply a different function that satisfies the above constraints, and also completely changes the vector values: for each embedding in a dataset, we generate a new, random vector of the same dimensionality. We then scale the new dimension values to match the norm of the original vector. This completely replaces the dimension values

with meaningless noise, invalidating any semantics assigned to a particular dimension, but specifically retains the norm values from the original vectors.

3.3.2 Ablating the Norm Container

As noted by Goldberg (2017), normalising vectors removes word frequency information, so presumably normalising all vectors in the dataset would also remove any information encoded in the norm. Normalisation equalises the magnitude of the vectors by scaling the values in each vector's dimensions in such a way that all vectors end up having the same norm, yet the dimensions' relative sizes remain unchanged.

However, we are conscious that vectors have more than one kind of norm, and can thus be normalised in different ways. Hence, we would need to choose a normalisation algorithm to match the norm that we wish to ablate, as a vector can only be normalised according to one norm at a time (e.g. either L1 or L2, not both). Unfortunately, given that there is a very high correlation between information in both norms[3], this means that if we perform an L1 normalisation, the information encoded in the L2 norm might remain, meaning the vector's norm information will not be completely removed[4].

Instead, we can apply a noising function analogous to the dimension ablation function: for each embedding in the dataset we generate a random norm value, then scale the vector's original dimension values to match the new norm. This randomises vector magnitudes, while the relative sizes of the dimensions remain unchanged. In other words, all vectors will keep pointing in the same directions, and the angle between any two vectors will remain the same—vectors that would be considered similar to each other in this way will continue to be

[3]The Pearson correlation coefficient between the L1 and L2 norms ranges between 0.96 and 0.97 on the different datasets used in later chapters of this thesis, showing very high correlation. Still, the correlation does not equal 1, also indicating there is a slight difference in the information encoded by the two norms.

[4]In a brief supplementary analysis we have found that normalising to one of the norms indeed removes the information encoded in that respective norm, but retains, or in some cases even amplifies the information in the remaining norm, indicating that this is not a viable way to ablate information from the norm information container. We present these supplementary results in Appendix A, see Table A.1.

similar after normalisation. However, any information encoded by differences in magnitude will be removed and replaced with random noise. In short, this function removes information potentially carried by a vector's norm, while still retaining dimension information[5].

3.3.3 Ablating Both Containers

Notably, these two approaches are not mutually exclusive, and as such can be combined. The vectors can be modified with both noising functions—ablating both the norm and dimensions should have a compounding effect, in essence sabotaging both information containers. In theory, we expect this would ablate all information encoded in the vector, as it essentially generates a completely random vector with none of the original information. As such, any probe trained on these vectors would have nothing to learn from and should preform comparably to random baselines.

Not only is it compelling to explore whether this would actually happen, but is actually a necessary step in the method, as it can confirm the ablative effect of the noising functions and allows us to check for redundancies between them. Indeed, our noise injections are meant to be interpreted sequentially and ablating both containers after ablating each one individually also acts as a sanity check that can inform our inferences. We illustrate this in Section 3.7.

3.4 Random Baselines

Even when no information is encoded in an embedding, a probe can learn the distributions of data and labels, especially if the train set contains class imbalance. There is also the possibility of a powerful probe detecting an empty signal (Zhang and Bowman, 2018; Hewitt and Liang,

[5]Given that vectors have more than one kind of norm, choosing which norm to scale to might not be inconsequential. We have explored this in additional experiments and found that in our framework there is no significant difference between scaling to the L1 norm vs. L2 norm. In other words, applying our norm ablation noise function to scale to the L2 norm removes information from both norms (evidence of this is presented as part of a post hoc analysis in Chapter 7, Table 7.7, as well as in Appendix A, Table A.1).

2019). We need to account for these variations as our method relies on relative evaluations, so we need to be able to account for these possible differences. To this end, we establish informative random baselines against which we can compare the probe's performance.

We employ two flavours of random baselines: (a) we assert a random prediction onto the test set, negating any information that a classifier could have learned, class distributions included; and (b) we train the probe on randomly generated vectors, establishing a baseline with access only to class distributions.

3.5 Confidence Intervals

Generally, recent work has called for greater rigour in evaluation approaches in NLP (McCoy et al., 2020; Sadeqi Azer et al., 2020), advocating for more widespread use of statistical tests and estimating the statistical power that tests on common benchmarks can provide (Card et al., 2020). With this in mind, we must account for the degrees of randomness in our method, which stem from two sources: (1) the probe may contain a stochastic component, e.g. a random weight initialisation; (2) the noise functions are highly stochastic (i.e. sampling random norm/dimension values). Due to this, evaluation scores will differ to varying degrees each time the probe is trained, making relative comparisons of scores problematic. To mitigate this, we retrain and evaluate each probing model a multitude of times (the total number of runs depending on dataset size and likely degrees of randomness) and report the average evaluation score of all runs, essentially bootstrapping over the random seeds.

However, when comparing mean scores of different models there might still be minor differences. In order to obtain statistical significance for the differences in averages, we calculate the confidence interval (CI). The CI provides a range of estimates for the true mean of a population, centred on the sample mean, and is defined as an interval with a lower bound and an upper bound. The interval is computed at a designated confidence level: while the 95% confidence level is most common, we opt for the 99% confidence level. The confidence

level represents the long-run frequency of confidence intervals that contain the true value of the parameter. In other words, 99% of confidence intervals computed at the 99% confidence level contain the true population mean.

Given a sample mean value m, the sample standard deviation σ and the sample size n, the confidence interval is defined by the following equation:

$$CI = m \pm Z\frac{\sigma}{\sqrt{n}} \tag{3.1}$$

where Z is the critical value, which depends on the desired confidence level, e.g. for a 99% confidence level it is 2.576, as provided by a Z table[6]. Note that the factors affecting the width of the CI include the confidence level, the sample size, and the variability in the sample. Larger samples produce narrower confidence intervals when all other factors are equal. Greater variability in the sample produces wider confidence intervals when all other factors are equal. A higher confidence level produces wider confidence intervals when all other factors are equal.

Thus, calculating the CI for a single mean will provide a range within which the true mean can be found. When comparing multiple means with a hypothesis that they might belong to different distributions, their CIs can provide statistical significance by confirming that observed differences in the averages of different model scores are significant. In practice, when comparing evaluation scores of probes on any two noise models, we use the CI range to determine whether they come from the same distribution: if there is overlap in the CI of two averages they might belong to the same distribution and there is no statistically significant difference between them. Using CIs in this way gives us a clearly defined decision criterion on whether any compared model performances are different. It also controls for dataset size,

[6]Z tables differ on usage, but essentially, the table tells us what the critical value is for many common probabilities. An example in the context of confidence intervals can be found here: `https://www.mathsisfun.com/data/confidence-interval.html`

meaning that this relative approach can work across different datasets, making comparisons between small and large datasets more principled.

3.6 Comparison to Other Methods

Our method accounts for the following criticisms of the probing framework, which we have presented in Section 2.3.2, as well as the introduction to the current chapter: (a) the need for intrinsic evaluations and (b) relative interpretations of results, (c) a grounding in statistical methods and (d) an emphasis on the importance of the norm, offering geometric insights into how embeddings store linguistic information. Some of these criticisms have already been raised by the community and efforts have been made to address them.

Intrinsic Evaluation: Torroba Hennigen et al. (2020) highlight the need for an intrinsic probe of embedding models and propose a novel framework based on a decomposable multivariate Gaussian probe that allows them to determine whether the linguistic information encoded in the dimensions of word embeddings is dispersed or focal. In contrast, our method focuses on the role of the vector norm, rather than dimensions, and can provide relevant intrinsic insights into the structure of an embedding model.

Relative Interpretation: Feder et al. (2020) construct counterfactual representations in order to compare the performance of the probe with and without the pertinent information, showing that by carefully choosing auxiliary adversarial pre-training tasks, language representation models such as BERT can effectively learn a counterfactual representation for a given concept of interest, and be used to estimate its true causal effect on model performance. In a related fashion, Elazar et al. (2020) directly remove relevant information from the representation, which allows for a comparison of probe performance with and without the removed information, in turn allowing them to measure the effects of confounding factors.

In essence, these efforts address the issue of relativising probe interpretations by removing information from the encoding; in that sense, our work finds its place alongside them.

However, our method is not designed to remove specific, pre-defined information, but is rather more exploratory in nature, with a focus on understanding where within an embedding certain information is encoded, achieved by a targeted disruption of embedding components. This type of analysis can improve our understanding of how an embedding encodes information, and, potentially, thereby provide insight into the signals within language that the embedding models use to recognise the presence of linguistic phenomena.

Statistical Method: the findings provided by our method are contingent on the use of confidence intervals and random baselines in order to establish statistical significance of results, which is not the focus of any existing methods. This gives us a way to claim statistically significant differences in evaluation results, offering a more principled basis for result interpretation. Our method thereby combines calls for more widespread use of statistical tests in NLP evaluation approaches (McCoy et al., 2020; Sadeqi Azer et al., 2020; Card et al., 2020) with findings on the importance of including baseline representations (Zhang and Bowman, 2018; Hewitt and Liang, 2019). Furthermore, our method opens the door for targeted post hoc statistical and experimental analyses, thereby offering a reframing of work in the related literature (we expand on this in Section 3.8).

Geometric Insights: in terms of obtaining insights into the structure of the representation space and the role of the norm, work by Hewitt and Manning (2019) is arguably most closely related to ours. The essence of their work is an investigation of structural properties of the word representation space, which finds that the norm plays a significant role in encoding information. Specifically, they designed a structural probe for finding syntax in word representations and performed experiments that provide insights into how a low-rank transformation recovers parse tree information from ELMo and BERT representations, finding that the depth of a sentence's parse tree is encoded by the vector norm.

In other words, their structural probe tests the concrete claim that there exists an inner product on the representation space whose squared distance—a global property of the

space—encodes syntax tree distance. This can be interpreted as finding the part of the representation space that is used to encode syntax. Given that they make a claim about the structure of the representation space and the role of the norm within it, our work is complementary to theirs.

One of the key differences between our work and that of Hewitt and Manning (2019) was expressed well by Elazar et al. (2020), whose approach also intervenes on the representation layers, which contrasts with related work that focuses on intervening in the input space (Goyal et al., 2019; Kaushik et al., 2020), or in specific parameters (Vig et al., 2020). This makes their approach easier than changing the input (which is non-trivial) and more efficient than querying millions of parameters. This observation holds for our method as well, given that our noise injections are applied at the representation layer. Additionally, this differentiates our method from the work of Hewitt and Manning (2019), as they do not intervene at the representation level, but rather at the probe level: they design a structural probe which is trained to recreate the syntax tree distance between all pairs of words in all sentences in the training set of a parsed corpus. Their probe also seems to only be applicable to contextual embeddings, as their findings depend on having varying word representations for different contexts.

3.7 Experiment Interpretation Guide

Having thus developed the method, we walk through the expected result interpretation process. Given that the method relies on a relative intrinsic comparison of different versions of a model, there will always be a number of results to consider, which might seem daunting at first glance, especially when applying the method to multiple models at a time. To help progress the results discussions in the later chapters, here we present a hypothetical example to serve as a basis for our experimental results interpretation. These faux-results are presented in Table 3.1.

For each model evaluation there will be 6 results presented: rows 1 and 2 will contain results of the two random baselines (random prediction and random vector respectively), while row 3 will contain the vanilla baseline result. After applying the various ablations and obtaining evaluations for the three ablated models, these results will be presented in rows 4, 5 and 6, respectively as ablated norm, ablated dimensions, and ablated both norm and dimensions.

For each model evaluation there will be two result columns, one presenting the classifier's average evaluation score (in this case accuracy, but it could be any other metric), and the other presenting the confidence interval (CI) for the average of all the training runs. As these will be average performance scores, we will use CI to establish statistically significant differences. If the interval ranges do not overlap this means they belong to different distributions, indicating that different amounts of information have been lost.

Cells will be shaded to indicate statistically significant differences in results: light grey if they belong to the same distribution as random baselines (i.e. no statistically significant difference from random); dark grey if they belong to the same distribution as vanilla baseline (i.e. no statistically significant difference from vanilla); and unshaded cells will contain scores that are significantly different from both the random and vanilla baselines.

As this is an intrinsic evaluation, results are interpreted vertically from top to bottom. However, in this fictional example, we will be examining four different models (M-1, M-2, M-3 and M-4) side by side, and their faux-results will illustrate the different conclusions we can draw from them. M-1 and M-2 will illustrate scenarios where no pertinent information is found in the norm, while M-3 and M-4 will show two scenarios that indicate that some pertinent information is stored in the norm.

M-1: We first examine the faux-results of model M-1. The random baselines establish a bottom performance below which no other model should be dropping. Meanwhile, unless the embeddings do not store any task-relevant information whatsoever, we expect the vanilla

row	model	M-1		M-2		M-3		M-4	
		ACC	±CI	ACC	±CI	ACC	±CI	ACC	±CI
1	rand. pred.	.5000	.0015	.5004	.0019	.4997	.0009	.5002	.0008
2	rand. vec.	.5001	.0012	.4995	.0025	.5005	.0011	.4997	.0016
3	vanilla	.8561	.0027	.8789	.0022	.9153	.0031	.8988	.0028
4	abl. N	.8555	.0035	.8667	.0018	.8967	.0027	.8978	.0021
5	abl. D	.5011	.0028	.5001	.0008	.5314	.0017	.5402	.0022
6	abl. D+N	.4998	.0015	.4989	.0025	.5002	.0018	.4999	.0017

Table 3.1 Hypothetical experimental results for four different embedding models evaluated with the probing with noise method. Reporting fictional average accuracy scores (ACC) and confidence intervals (CI) of the average accuracy of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

model to significantly outperform the random baselines. We would also expect it to outperform the ablations, as the essence of the method is a systematic removal of information from the representations, so all ablation scores should be lower than vanilla[7].

After establishing the baselines, given that our focus is to gain insights on how information is stored in the norm, we first look at norm ablations. While this result is slightly lower, it shows no statistically significant difference compared to vanilla. Ablating the norm from this representation does not decrease the score, which would indicate that no task-relevant information has been removed.

Certainly with our current understanding of language embeddings, we know that dimensions typically encode the bulk of the information, hence we would expect that in a setup where dimensions are ablated and no information is encoded in the norm, the model will significantly underperform when compared to the vanilla baseline. In this case, ablating the dimensions drops the scores quite low and makes them comparable to the random baselines.

Finally, ablating both norms and dimensions also causes a performance drop which makes the score comparable to random, indicating no difference between this setting or the pure dimension ablation setting. We consider this to be the prototypical scenario which shows

[7]We assume that these baseline considerations hold for all 4 models.

that absolutely no task-relevant information is encoded in the norm, as all results indicate that all the relevant information is stored in the dimension container.

M-2: The model M-2 scenario is slightly different—when just ablating the norm, the performance drop is statistically significant when compared to the vanilla baseline. However, when just the dimensions are ablated the performance immediately becomes comparable to random baselines and does cause a further significant drop when ablating both norm and dimensions.

Given our understanding of the underlying mechanics, we do not consider this sufficient evidence that the norm encodes the relevant information: even though the norm ablation causing a performance drop should indicate that some relevant information has been removed, if that were true, then having only that information available in the dimension ablation setting should yield above random performance. Seeing as it does not, the evidence for the norm's role is inconclusive; we suspect it is more likely an indicator of an interaction between the encoding and the noise function, or perhaps of some kind of interdependence of information between the norm and dimensions—the information in the norm supplements the dimension information for an increased performance score, but on its own is not sufficiently informative to beat random scores.

M-3: On the flip side, we consider the faux-results of model M-3 as the prototypical example that indicates there is at least some information encoded in the norm: when performing a norm ablation, the score drops significantly compared to vanilla. When performing a dimension ablation, the score drops further, but remains above random performance. When both norm and dimensions are ablated together, the score is not different from random baseline performance.

This is a strong indicator that the norm encodes some information independently from the dimensions: even with absolutely no dimension information, the probe can still learn

some information relevant to the task just based on the vector norms. The representation only reaches a state of no relevant information when both norm and dimensions are ablated.

M-4: Model M-4 differs from M-3 as just ablating the norm does not cause a significant performance drop compared to vanilla. However, this does not necessarily indicate that the norm does not contain any task-relevant information at all—looking at M-4 with ablated dimensions shows that it still outperforms random baselines. This means that the norm on its own is sufficient to help the probe solve the task, indicating that it does carry some relevant information.

Somewhat analogously to M-2, where we suspect a kind of information supplementation between the norm and dimensions, in M-4 we suspect the lack of a significant drop when ablating the norm to be due to some kind of information redundancy between the norm and dimensions: the information in the norm could also be present in the dimensions, so when only the norm is ablated we observe no performance drop, as no information was lost. However, ablating the dimensions removes most of the information from the embedding, but the probe can still learn some task-relevant information from the residual information in the norm. While in the analogous scenario we consider model M-2 as not providing sufficient evidence that the norm encodes relevant information, we consider that the results in scenario M-4 do. Though the information might be redundant with the dimensions, this result still demonstrates that the norm is capable of encoding information regardless of what is in the dimensions.

Generally, models M-2 and M-4 show that just ablating the norm is not necessarily sufficient to establish whether the norm encodes task-relevant information or not. As a rule of thumb, we can say that the most important indicator of the norm's importance is the comparison of results in row 5 and row 6 (and random baselines): if for a particular task performance remains above random after ablating dimensions, but drops to random when

ablating both dimensions and norms, this is a strong indicator that the norm is encoding the relevant information.

These are only fictional results, but they illustrate the type of insights that our method can provide. Certainly, our real experimental results might not always be as straightforward as these, but we consider them as the general guiding principle when interpreting our probing results.

3.8 Post Hoc Analyses

As exemplified by the above hypothetical scenarios, our method allows us to discriminate whether the information is encoded in the norm, dimensions or both. It can decouple the two information storage containers and in doing so opens the door for further, more specific insights. Having knowledge of which container encodes the relevant information allows us to perform additional targeted experiments or statistical tests that can deepen our understanding of *how* the information is encoded in a particular container.

This is akin to the way post hoc tests are sometimes applied in statistics. For example, post hoc tests are an integral part of ANOVA: when using ANOVA to test the equality of at least three group means, obtaining statistically significant results only indicates that not all of the group means are equal, but it does not identify which particular differences between pairs of means are significant. This can only be revealed by using post hoc tests to explore differences between multiple group means.

In the case of our method, once we know which information container encodes the relevant information, we can follow up on our line of questioning by trying to understand how the information is encoded in a given container. Such post hoc testing can then be done either on the dimension container, or the norm container. In order to better understand where in the dimensions relevant information is encoded, the post hoc analysis can include techniques such as principal component analysis, performing additional dimension ablation

experiments or any approaches related to the work done by Torroba Hennigen et al. (2020) and others. Meanwhile, post-hoc analysis on the norm container can include correlation studies between the norm and other vector features, or a study of the norm's correlation with the class labels, among other examples.

Certainly, this raises the question of why our method is necessary, when we could potentially gain the same insights by, for example, just performing a correlation study. To illustrate why this would be insufficient, we offer another hypothetical example: given a dataset of vector representations with assigned class labels, we wish to find out whether the vector norm encodes some of the relevant information. To test this, we perform a correlation analysis between the norm values and the class labels. This can give us one of two results: (a) it can reveal that there is no correlation between the class labels and the norm, or (b), as long as the correlation coefficient is non-zero, this could be taken as an indicator that some amount of relevant information is encoded in the norm[8].

We cannot take finding (a) as definitive proof that there is no information in the norm. In part, this is because many typical correlation coefficients test for a linear relationship, while the relationship between the two variables might be non-linear and would be more aptly represented by a non-linear model. Certainly, this could be avoided by employing the correct tests, however such tests are not common and the choice of the correct test is not trivial, which still leaves us with the risk of obtaining a false negative result.

Furthermore, by only relying on a correlation coefficient, we also run the risk of (b), a false positive result. Even if some non-zero correlation is detected, this is too weak of a signal to be considered definitive, as confounding factors could be at play and the relationship between the norm and labels might be spurious[9].

In the case where there are no confounders, this still leaves room for an occurrence similar to the M-2 scenario laid out in the previous section: in M-2 the information encoded

[8] Whether the correlation is weak or strong is not relevant in this example.

[9] Indeed, we will encounter such examples on real data, and discuss this in Chapter 7.

in the norm is only complementary with dimension information, supplementing it for an increased performance score, but on its own is not sufficiently informative to beat random scores. We cannot reach this sort of conclusion by studying the norm in isolation. In fact, as a more general principle, in the previous section we have established that ablating the norm itself cannot provide a clear-cut answer. We thus posit that an analysis that does not rely on our full method cannot provide a complete picture of this relationship.

However, when used in conjunction with our method, it can provide valuable additional insight into the way information is encoded in an information container. For example, the correlation coefficient being positive or negative can reveal the relative distance of the vectors in question from the origin of the space. Similarly, the right post hoc analysis of the dimension container can reveal whether information is localised or distributed across dimensions, or can perhaps reveal which subset of dimensions is relevant for encoding a given linguistic property.

That said, we do not insist that any post hoc analysis is a necessary step in our method, nor do we prescribe the type of tests that should be done, as this is contingent on the research interests. Rather, we simply highlight that such post-hoc tests can be done but leave the choice up to the researcher.

More importantly, positioning such experiments as post hoc analyses offers a slight reframing of the way we think about prior work in this space, most of which makes an implicit assumption that the information is either in dimensions (most often) or norm, without first testing this assumption. Yet a necessary prerequisite for doing any embedding analysis is to affirm this presupposition that whichever container is being tested or experimented on is the only relevant source of information, and if not, making considerations about how the applied analysis impacts the remaining container. Our method provides this kind of insight and allows for a more principled application of such tests and experiments. We demonstrate some post hoc experiments and analyses in Chapters 5 and 7.

Having established the core method, we apply it to a range of linguistic probing tasks. We begin with a taxonomic hypernym-hyponym classification probing task in Chapter 5. However, in order to probe encodings of taxonomic information, we must first create them. We describe the creation of our taxonomic representations in the following Chapter 4, and then probe them in Chapter 5.

Chapter 4

Creating Taxonomic Representations

One of the aims of this thesis is to apply the *probing with noise* method to different kinds of embeddings on a large variety of probing tasks to see which linguistic information, if any, is encoded in their norm. We begin this exploration with taxonomic embeddings: they are particularly interesting for this application as we suspect that the hierarchical structure of a taxonomy is well suited to be encoded by the vector norm—given that the norm encodes the vector's magnitude, or distance from the space's origin, it is possible that the depth of a tree structure, such as a taxonomy, could be mapped to the vector's distance from the origin in some way. Applying our method to taxonomic embeddings on a taxonomic probing task could shed some light on this relationship.

Probing word embeddings for taxonomic information naturally requires taxonomic word embeddings. Rather than use pretrained taxonomic embeddings, which are not commonly available, we instead elected to train our own ones. Upon considering the available options (see Section 4.1), we settled on using the random walk algorithm over the WordNet taxonomy, inspired by the work of Goikoetxea et al. (2015). We made this choice as it allows us to be methodologically consistent, making our model and result comparisons less prone to certain types of confounders; it allows us to create taxonomic embeddings while using the same architectures used to obtain thematic embeddings (more on this in Sections 4.1 and 4.5).

In short, the approach is to generate a pseudo-corpus by crawling the WordNet structure and outputting the lexical items in the nodes visited, and then running the word embedding training on the generated pseudo-corpus. Naturally, the shape of the underlying knowledge graph (in terms of node connectivity: i.e. tree, fully-connected, radial etc.) affects the properties of the generated pseudo-corpus, while the types of connections that are traversed will affect the kinds of relations that are encoded in this resource. Developing a better understanding of the relationship between the shape of a knowledge graph, the properties of the resulting pseudo-corpora, and the properties of the resulting embeddings, has the potential to inform how the walk over a given knowledge graph should be tailored to improve taxonomic encodings, and will help us decide how to best generate the embeddings for our taxonomic probing task.

This chapter describes in detail the creation of our WordNet random walk taxonomic word embeddings and the accompanying evaluation of the embeddings on the task of word similarity. Note that we do not yet apply our *probing with noise* method to evaluate the taxonomic embeddings in this chapter. Rather, we first validate that they have been correctly generated and that they encode taxonomic information at all: we do this by applying existing evaluation frameworks which allow for comparability to related work on taxonomic embeddings. We also need to examine the properties of the generated pseudo-corpora in order to obtain a better understanding of the impact of their features and possible confounding factors on the resulting embeddings. Once the resources have been understood and the validity of the embeddings has been established, we move on to applying the *probing with noise* method in Chapter 5.

Finally, as outlined in Chapter 1, while a number of publications have arisen from the work done on this chapter, here we only present results published in first-author papers (Klubička et al., 2019; Klubička et al., 2020), and do not present the sister-publication on retrofitting taxonomic word embeddings (Maldonado et al., 2019) beyond the overlap in

related work, as our involvement in those experiments was more collaborative and the results are outside the scope of this thesis.

4.1 Taxonomic Representations

Research on building embeddings from knowledge resources can be broadly categorised into three approaches: (1) **taxonomic enrichment** approaches that seek to augment the similarity of words in pretrained embeddings, based on their taxonomic relationship as expressed by a knowledge resource (this is in addition to the thematic relations already learned through their original corpus training), (2) **semantic specialisation** techniques that modify pretrained vectors in such a way so that their cosine similarity ends up measuring a specific semantic relation, and (3) **knowledge-resource encoding** methods that directly learn knowledge resources.

Both enrichment and specialisation modify pre-computed, corpus-based word embeddings with information from a knowledge resource to either augment them (enrichment) or to fit them onto the specific semantic relation described by that knowledge resource (specialisation). Retrofitting (Faruqui et al., 2015) is an example of enrichment: it modifies corpus-based embeddings by reducing the distance between words that are directly linked in resources like WordNet (Fellbaum, 1998), MeSH (Yu et al., 2016) and ConceptNet (Speer and Havasi, 2012). In our own related work, we have explored the impact of corpus size on vector enrichment (Maldonado et al., 2019).

On the other hand, specialisation involves fitting pre-computed corpus-based word embeddings onto a specific semantic relation described by a knowledge resource. Examples of this include PARAGRAM (Wieting et al., 2015), Attract-Repel (also called counter-fitting) (Mrkšić et al., 2016), Hypervec (Nguyen et al., 2017), as well as the work of Nguyen et al. (2016) and Mrkšić et al. (2017) on synonyms and antonyms. By applying different modifications to the objective function, the aim of such research is to convert the cosine similarity into

a function that measures the specific type of semantic relation that is learned, while weighting down the thematic relationship originally learnt during pretraining on a text corpus. More recently, Vulić et al. (2018) and Ponti et al. (2018) have introduced global specialisation models where vectors for words that are missing in the knowledge resource are also updated.

Our work is more related to approaches to learn directly from knowledge resources. An example of this is creating non-distributional sparse word vectors from lexical resources (Faruqui and Dyer, 2015), with each dimension representing whether a word belongs to a particular synset, holds a particular taxonomic relation, etc. According to Hamilton et al. (2017), to embed a graph is to learn a vector representation of each node such that geometry in the vector space—distances and norms—approximates geometry in the graph: examples of this include building Poincaré embeddings that represent the structure of the WordNet taxonomy (Nickel and Kiela, 2017), and building embeddings that encode all semantic relationships expressed in a biomedical ontology within a single vector space (Cohen and Widdows, 2017). These two methods encode the semantic structure of a knowledge resource in a deterministic manner, while Agirre et al. (2010) follow a stochastic approach based on Personalised PageRank: they compute the probability of reaching a synset from a target word, following a random-walk on a given WordNet relation.

Instead of computing random-walk probabilities, Goikoetxea et al. (2015) use an off-the-shelf implementation of the word2vec Skip-Gram algorithm to train embeddings on WordNet random walk pseudo-corpora, changing neither the embedding algorithm nor the objective function[1]. The resulting embeddings encode WordNet taxonomic information rather than natural word co-occurence. A characteristic of WordNet random-walk embeddings is that they are of the same "kind" as typical word embeddings, in the sense that both are distributional and are trained to satisfy the same objective function. If settings and hyperparameters are kept the same, as far as the embedding model is concerned, the only difference between the two sets of vectors is that they were trained on different corpora. As such, this gives them

[1] `http://ixa2.si.ehu.eus/ukb/`

the advantage that they can either be used as is, or can be combined with natural-corpus embeddings in order to accomplish enrichment or specialisation (Goikoetxea et al., 2016; Maldonado et al., 2019). Still, it is important to note that the contexts for target words in both embedding types are categorically different: contexts in natural text are made of naturally co-ocurring words, reflecting non-taxonomic and thematic relationships. In contrast, contexts in WordNet random-walks are words that are taxonomically related to the target word.

Finally, Simov et al. (2017b) build directly on the work of Goikoetxea et al. (2015) and explore how various different varieties of the random walk algorithm impact performance of trained word embeddings, similar to our own work on the topic (Klubička et al., 2019). They pour significant effort into techniques for enriching WordNet's graph structure and populating it with as many additional semantic connections as possible (Simov et al., 2015, 2016a,b), leveraging all available relationships between WordNet synsets, as well as adding and inferring more from external resources (Simov et al., 2017a,b).

4.1.1 Evaluation Benchmarks

The quality of vectors produced by knowledge-resource encoding, semantic specialisation and taxonomic enrichment have been evaluated through diverse semantic similarity benchmarks. These benchmarks include WordSim-353 (Finkelstein et al., 2002), which conflates taxonomic similarity with thematic similarity; SimLex-999 (Hill et al., 2015) which focuses on taxonomic similarity; and SemEval-17 (Camacho-Collados et al., 2017), which considers thematic and taxonomic similarity as two points on a scale of degrees of similarity. See Section 4.5 for more details on these benchmarks.

Table 4.1 shows Spearman correlation scores on WordSim-353, SimLex-999 and SemEval-17 of example systems from the literature that implement the three approach families mentioned earlier. In general, performance tends to be worse on SimLex-999 than on SemEval-17 and WordSim-353. However, notice that Attract-Repel (Mrkšić et al., 2017) has obtained

scores as high as 0.71 on SimLex-999, likely as it specialises in learning (and distinguishing between) synonymic and antonymic relations and incorporates information from rich knowledge sources.

Of special note in these results is that Goikoetxea et al. (2016) found that simple vector concatenation (RW+SG in Table 4.1) performs better than retrofitting (and other more complex methods of vector combination) on WordSim-353 and SimLex-999. The original retrofitting method Faruqui et al. (2015) used the Paraphrase Database (Ganitkevitch et al., 2013), WordNet and FrameNet (Baker et al., 1998) ontologies. They achieve a Spearman score of 0.70 on the WordSim-353 dataset. However, their work is focused only on using synonyms derived from synsets, and they do not make use of other types of relations found in knowledge bases, such as hypernymy and hyponymy.

The original winners of the SemEval-17 competition employed retrofitting in their system (Speer and Lowry-Duda, 2017). They perform what they call "expanded retrofitting", which means that they use a union of the vocabularies from the corpus embeddings and semantic network, as opposed to regular retrofitting where the vocabularies are intersected. In addition, they use ConceptNet (Speer and Havasi, 2012) instead of WordNet, and employ heuristics to handle out-of-vocabulary words, such as averaging the vectors of the neighbours of a given out-of-vocabulary word in the semantic network. With this system, they achieve a Spearman score of 0.80 (Table 4.1).

Despite the appealing simplicity and strong performance of the embeddings resulting from the concatenation of random-walk and natural corpus embeddings (RW+SG in Table 4.1), they have received little attention in the literature. One exception is our sister-experiments (Maldonado et al., 2019), where we set up a vector enrichment scenario and performed and in-depth exploration of how the relative sizes of the thematic and taxonomic corpora used to train embeddings affect the performance of the resulting representations. This is an important consideration as typically the quality of vectors increases in proportion to the size of training

Method Type	Method	Ref.	WS	SL	SE
Text	SG	Goikoetxea et al. (2015)	.69	.44	.57*
Encoding	PPR/WN	Agirre et al. (2010)	.72	--	--
Encoding	RW/WN	Goikoetxea et al. (2015)	.70*	.52	.50*
Enrichment	RW+SG	Goikoetxea et al. (2015)	**.80**	.55	.72*
Enrichment	Retrofitting	Faruqui et al. (2015)	.70	.44*	**.80****
Specialisation	Attract-Repel	Mrkšić et al. (2017)	--	**.71**	--

Table 4.1 Spearman scores of a selection of methods on three benchmarks: WordSim-353 (WS), SimLex-999 (SL) and SemEval-2017 (SE). Highest value in each benchmark column is state of the art for that benchmark. Abbreviated methods are:
SG: text embeddings trained via Skip-Gram.
PPR/WN: Personalised Page-Rank over WordNet.
RW/WN: Random-Walk over WordNet.
RW+SG: RW/WN vectors concatenated to SG vectors.
* Evaluated in our sister experiments (Maldonado et al., 2019).
** Evaluated by Speer and Lowry-Duda (2017) in their experimental reproduction.

data, yet given that the WordNet structure is finite, doing very extensive random walks, potentially revisiting the full structure more than once and thus overfitting over the topology of the knowledge graph, may not actually be so beneficial. Our results have shown that there is a "sweet spot" in terms of adding more taxonomic data versus more natural corpus data: taxonomic enrichment does not always improve the performance of the embeddings, and where performance does increase, only medium sizes of random walk corpora are required, i.e. in an enrichment scenario there is little benefit to training vectors on very large random walks.

However, even with these findings, there has been no work on analysing the properties of the corpora generated by random-walk processes. In particular, there has been no work on comparing their statistical properties with those of natural corpora, nor a study on the impact of confounding factors on the performance of the resulting embeddings. We address these questions as part of the embedding validation process in this thesis, and the results of the work have been published in related venues (Klubička et al., 2019; Klubička et al., 2020). Additionally, with the recent prominence of probing techniques, it seems very few have been

applied to any kinds of taxonomic embeddings, so there is untapped potential in applying the probing framework to these embeddings as well, in addition to the usual thematic ones. After generating our WordNet random walk taxonomic embeddings, we apply our *probing with noise* method to them and perform an intrinsic evaluation in Chapter 5.

4.2 Random walk pseudo-corpus generation

Our pseudo-corpus generation process is inspired by the work of Goikoetxea et al. (2015). The core idea of the corpus generation algorithm is that it generates a 'sentence' by performing a random walk over the taxonomic graph of WordNet (Fellbaum, 1998). By randomly walking the WordNet knowledge graph and choosing words from each synset that has been traversed, a pseudo-corpus is generated and used for training word embeddings, in the same way one would train on a natural language corpus. The reasoning behind this approach is that the distributional hypothesis should also apply in this scenario, in the sense that co-occurrence within local contexts in the pseudo-corpus will reflect the connections between words connected in the WordNet graph. In other words, using this approach flattens out the WordNet taxonomy, turning it into a sequential format similar to a natural corpus, where the same implicit connection, i.e. co-occurrence, reflects taxonomic relations, rather than thematic ones.

A random walk begins at a randomly selected synset in the WordNet graph and randomly moves to an adjacent synset. Each time the walk reaches a synset, a lemma belonging to the synset is emitted. When the random walk terminates, the sequence of emitted words forms a pseudo-sentence of the pseudo-corpus. This process repeats until a predetermined number of sentences have been generated.

We use three hyperparameters to control the random walk over the graph: (i) a dampening hyperparamter α, (ii) a directionality hyperparameter, and (iii) a minimum sentence length hyperparameter.

(i) The dampening factor (α) is used to determine when to stop the walk, so that at each step the walk might move on to a neighbouring synset with probability (α), or might terminate with the probability ($1 - \alpha$). Goikoetxea et al. also used a dampening factor and found the best practice is to set it to 0.85. We briefly experimented with slightly higher or lower values, but found it had relatively little impact on pseudo-sentence length when compared to the impact of the other hyperparameters, hence we set ours to 0.85 and did not change it further. While the dampening parameter was introduced by Goikoetxea et al., the directionality hyperparameter and the minimum sentence length hyperparameter represent extensions that we have introduced ourselves.

(ii) The directionality parameter constrains the permissible directions that the walk can proceed along as it traverses the taxonomic graph (e.g., only up, only down, both). We can do this because we exclusively traverse the WordNet taxonomy, i.e. we only consider hypernym/hyponym connections, which have an inherent directionality to them. This allows us to consider the graph's edges as directed, rather than, as Goikoetxea et al. did, treat them as undirected (due to considering a variety of connections that are not all directional). The motivation for introducing this hyperparameter is that it permits us to explore the relationship between variations in the random walk algorithm, variations in the shape of the underlying graph and the varying properties of the generated corpora.

(iii) The minimum sentence length parameter enables us to filter the sentences generated by the random walk algorithm by rejecting any sentence that is shorter than a prespecified length n. This allows us to explore the impact of different sentence lengths on the resulting corpora and embeddings, but also doubles as a filtering mechanism that allows us to filter out words which are not well connected to the taxonomy. Given that we allow our algorithm to start the random walk anywhere in the graph, if not for this constraint, the walk would often begin, and end, at a disconnected node. The taxonomic graph is quite sparse—if

we only walk along the taxonomic edges, a lot of nodes present in WordNet will end up disconnected, as some synsets are not part of the taxonomy, but are connected via other, non-taxonomic relations. If no minimal sentence length constraint is imposed, this leaves the synthesized pseudo-corpus containing many one-word pseudo-sentences, which are not informative in terms of their taxonomic relationships to other words. In this sense, minimal sentence length is a necessary hyperparameter if the goal is to constrain the vocabulary of the random walk pseudo-corpus to only the taxonomic graph of WordNet and discard all words that are not connected to it via a hypernym or hyponym relation. However, even if this is not a concern, the parameter also enables us to generate a corpus of sentences of any minimal length, allowing for a study of different pseudo-corpora properties.

More on all three hyperparameters is explained in Section 4.3.

Controlled by these hyperparameters our random walk algorithm progresses as follows: The random walk starts at a random synset and chooses a lemma corresponding to that synset based on the probabilities in the inverse dictionary (the mapping from synsets to lemmas) provided by WordNet. However, these are expressed as frequencies, rather than explicit probabilities, so we choose one based on the probability distribution derived from the frequency counts. Once the lemma has been emitted, the algorithm stochastically decides whether the walk should be terminated or not, controlled by the hyperparameter α. Terminating the walk determines the end of the pseudo-sentence, which is then added to the pseudo-corpus and a new random walk is initiated. If the walk is not terminated we check if the synset has any hypernym and/or hyponym connections assigned to it (depending on the direction constraint). If it does, we choose one at random with equal probability and continue the walk towards it, choosing a new lemma from the new synset. This process continues until one of two conditions are met: (a) the dampening factor (α) terminates the process, or (b) there are no more connections to take. We then restart the process and create a new pseudo-sentence. This pseudo-sentence generation process is repeated until we have

generated the required number of sentences. One important thing to note is that we allow our algorithm to go back to a node that has already been visited, but we do not allow it to choose a lemma that has already appeared in the sentence we are generating at the time.

While our pseudo-corpus generation process is based on the work of Goikoetxea et al. (2015), there are a number of important differences between the two algorithms. First, Goikoetxea et al. performed random walks over the full WordNet knowledge base as an undirected graph of interlinked synsets, making use of all available connections in the graph, whereas we only traverse the hypernym/hyponym relationship and ignore non-taxonomic relationship types such as gloss, meronym and antonym relations. This effectively allows us to traverse the taxonomic graph of WordNet exclusively. The main motivation behind this decision is that primarily, we are interested in embedding taxonomic relatedness from the generated corpus, and constraining the random walk to the taxonomic relationships is the most explicit way of doing so. This restriction to the taxonomic components of the graph has two important implications: (i) it permits us to consider the graph as directed (hypernym/hyponym→up/down), and (ii) it makes the full graph quite sparse. These implications have allowed us to further diverge from Goikoetxea et al.'s work and implement the directionality and minimal sentence length hyperparameters as described above. In addition, as opposed to Goikoetxea et al. who produce multiword terms, such as `Victrola_gramophone`, `telephone_call` and `shatterproof_glass` essentially treating them as words with spaces, in our corpora we divide these terms up into their individual constituent words (e.g. `Victrola gramophone`, `telephone call` and `shatterproof glass`). Though this is not the traditional approach to handle multi-word terms, we do so to make them more compatible for retrofitting with natural corpora, which we took advantage of in our related research (Maldonado et al., 2019)[2]. With that in mind, the following are examples of typical pseudo-sentences

[2]However, our implementation also allows for the option of generating pseudo-sentences where multi-word expressions are not split. It also allows generating sentences that include words found in synsets that are disconnected from the taxonomy, which results in better vocabulary coverage, but potentially poorer taxonomic representation. We make our implementation publicly available on GitHub (see Section 4.6)

that can be found in our pseudo-corpora, containing only words with taxonomic relations between them:

- *measure musical notation tonality minor mode*
- *decouple tell dissociate differentiate know distinguish*
- *vocalizer castrato vocaliser rapper vocalist caroler*
- *call-back call call-in telephone call trunk call*
- *meeting place facility station first-aid station aid station*

4.3 Pseudo-corpora properties

Using the approach outlined in Section 4.2, we generated taxonomic pseudo-corpora for the following combinations of hyperparameters:

1. **Size**. We define corpus size in terms of the number of pseudo-sentences generated. We generate pseudo-corpora of sizes 1k, 10k, 100k, 500k, 1m, 2m and 3m sentences.
2. **Direction**. As we are only walking the WordNet taxonomy, we define direction as allowing the walk to either only go up the hierarchy, down the hierarchy, or both ways.
3. **Minimum sentence length**. We impose a constraint on minimal sentence length and generate corpora with a 1-word, 2-word and 3-word minimum sentence length.

Combining these hyperparameters yielded a total of 63 pseudo-corpora of varying sizes, directions and minimal sentence lengths. Additionally, for the purpose of the taxonomic enhancement set of experiments (Maldonado et al., 2019), we also generated an additional 18 corpora without direction or sentence length constraints (i.e. allowing the walk to traverse both directions and allowing 1-word sentences). These additional corpora are much larger,

upwards of 468 million sentences. We have publicly released all of the generated corpora; however, due to the fact that the larger corpora were generated with constant hyperparameters, in this chapter we only discuss statistical data and analyses of the corpus groups of up to 3 million sentences. Furthermore, because the corpora that contain 1-word sentences by definition contain words found outside the taxonomic graph of WordNet, they are not strictly taxonomic and reflect a graph structure that is not a tree—a distinction that informs the discussion and analysis of our work. As such, they fall outside the scope of this chapter and we thus exclude corpora with 1-word sentences from the discussion in this section, as well as the evaluation in Section 4.5. Still, we have released them together with all other corpora (see Section 4.6), and their statistics are included in Tables 4.2 and 4.3.

Having generated WordNet random-walk corpora, before we use them to train embeddings, it is pertinent to examine their properties. This will allow us to establish a better understanding of their nature and the impact of possible confounding factors such as rare words. More generally, given that these are synthetic corpora used as training data for algorithms that were designed with natural corpora in mind, it would be wise to examine their statistical features and compare them to properties of natural corpora. Such insight could deepen our understanding of the resulting embeddings and could help inform the interpretation of our results.

Starting with descriptive statistics, for each generated pseudo-corpus we measure the following: total number of tokens, average sentence length (average tokens per sentence), percentage of identical sentences, size of vocabulary, and percentage of rare words in the vocabulary. This data is presented in Tables 4.2 and 4.3.

Token count and sentence length. From the tables it is clear that the total number of tokens grows with the size in terms of number of pseudo-sentences in a corpus. Interestingly, however, although the average sentence length correlates with absolute number of tokens, it stays constant regardless of the number of sentences, all other things being equal. For

size	direction	min.sent.len.	token count	avg.sent.len.	%same sents	vocabulary	%rare words
1k	up	1w/s	4,921	4.92	0.10	2189	84.74
1k	down	1w/s	1,603	1.60	0.50	1425	60.28
1k	both	1w/s	3,378	3.38	0.20	2540	88.62
1k	up	2w/s	7,013	7.01	0.00	2569	96.77
1k	down	2w/s	2,918	2.92	1.00	2280	99.91
1k	both	2w/s	4,691	4.69	0.00	3212	99.47
1k	up	3w/s	7,957	7.96	0.10	2621	96.26
1k	down	3w/s	4,216	4.22	1.70	2895	99.79
1k	both	3w/s	5,519	5.52	0.30	3671	99.48
10k	up	1w/s	48,990	4.90	1.90	12643	77.93
10k	down	1w/s	16,009	1.60	5.87	10810	55.62
10k	both	1w/s	35,085	3.51	2.13	16830	84.34
10k	up	2w/s	70,433	7.04	0.62	12929	93.74
10k	down	2w/s	29,537	2.95	7.18	13943	97.66
10k	both	2w/s	48,022	4.80	0.85	18972	96.37
10k	up	3w/s	80,351	8.04	0.62	13231	93.33
10k	down	3w/s	41,987	4.20	12.40	13857	94.41
10k	both	3w/s	55,988	5.60	0.43	21038	95.91
100k	up	1w/s	492,133	4.92	12.92	51900	68.49
100k	down	1w/s	159,533	1.60	33.03	51412	50.13
100k	both	1w/s	351,970	3.52	13.24	62699	74.28
100k	up	2w/s	705,977	7.06	5.30	44482	87.25
100k	down	2w/s	295,042	2.95	38.56	39999	83.49
100k	both	2w/s	479,014	4.79	6.57	56358	85.43
100k	up	3w/s	804,104	8.04	4.79	44899	86.89
100k	down	3w/s	419,782	4.20	45.70	33118	72.31
100k	both	3w/s	564,113	5.64	3.39	58743	83.68
500k	up	1w/s	2,459,643	4.92	31.66	84842	59.18
500k	down	1w/s	798,474	1.60	68.06	84727	48.95
500k	both	1w/s	1,761,568	3.52	32.71	88707	47.84
500k	up	2w/s	3,515,524	7.03	18.50	64,257	67.35
500k	down	2w/s	1,475,336	2.95	68.56	55,508	53.35
500k	both	2w/s	2,401,498	4.80	20.06	67,049	39.86
500k	up	3w/s	4,011,247	8.02	17.06	63,923	66.48
500k	down	3w/s	2,097,641	4.20	71.01	46,701	52.33
500k	both	3w/s	2,822,171	5.64	12.22	67,353	33.30

Table 4.2 Statistics of generated random walk pseudo-corpora ranging from 1k to 500k pseudo-sentences in size. Statistics are presented in groups based on hyperparameters: we first present size, then minimal sentence length, then direction. Rows presenting data on corpora with a 1-word sentence minimum are shaded cyan, 2-word sentence minimum are shaded magenta and 3-word sentence minimum are shaded orange.

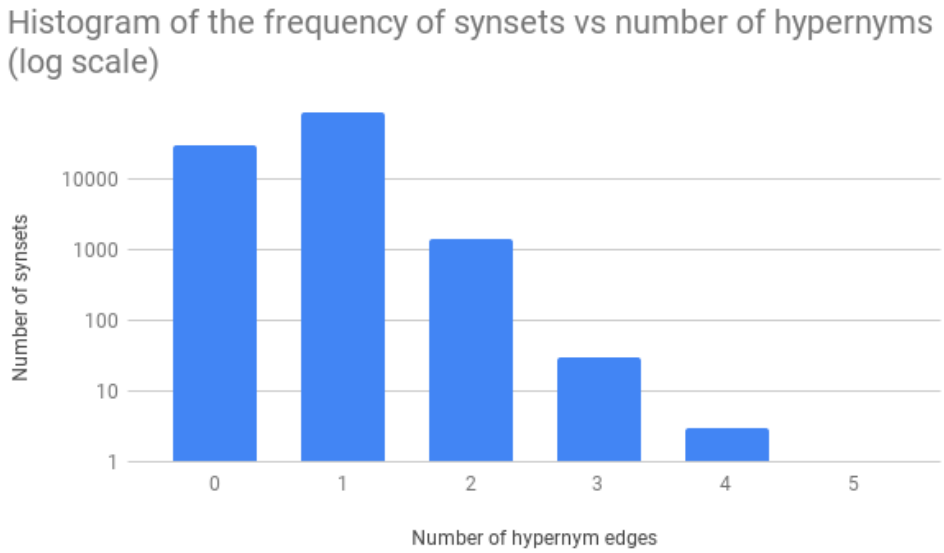
size	direction	min.sent.len.	token count	avg.sent.len.	%same sents	vocabulary	%rare words
1m	up	1w/s	4,924,245	4.92	41.38	90731	46.38
1m	down	1w/s	1,596,776	1.60	79.75	90494	43.93
1m	both	1w/s	3,515,489	3.52	42.32	91958	25.68
1m	up	2w/s	7,041,365	7.04	27.93	66,840	41.84
1m	down	2w/s	2,947,657	2.95	78.57	59,894	40.81
1m	both	2w/s	4,802,354	4.80	28.49	67,647	15.82
1m	up	3w/s	8,032,165	8.03	26.31	66,401	40.52
1m	down	3w/s	4,195,458	4.20	79.46	51,310	43.91
1m	both	3w/s	5,636,469	5.64	18.88	67,683	11.31
2m	up	1w/s	9,828,501	4.91	51.55	92773	25.68
2m	down	1w/s	3,195,186	1.60	87.63	92682	34.02
2m	both	1w/s	7,031,643	3.52	51.29	93119	9.92
2m	up	2w/s	14,079,962	7.04	39.56	67,587	19.32
2m	down	2w/s	5,898,583	2.95	85.91	63,089	30.03
2m	both	2w/s	9,602,490	4.80	37.66	67,756	3.88
2m	up	3w/s	16,061,599	8.03	37.65	67,081	18.20
2m	down	3w/s	8,389,396	4.19	85.92	55,314	35.99
2m	both	3w/s	11,274,757	5.64	26.99	67,757	2.34
3m	up	1w/s	14,767,000	4.92	57.37	93,187	15.32
3m	down	1w/s	4,790,103	1.60	90.78	93,140	27.18
3m	both	1w/s	10,554,177	3.52	56.17	93,366	4.35
3m	up	2w/s	21,131,926	7.04	46.67	67,714	9.48
3m	down	2w/s	8,849,429	2.95	89.16	64,416	24.56
3m	both	2w/s	14,402,423	4.80	43.00	67,772	1.41
3m	up	3w/s	24,084,882	8.03	44.78	67,198	8.93
3m	down	3w/s	12,580,624	4.19	88.89	57,499	31.67
3m	both	3w/s	16,918,222	5.64	32.14	67,776	0.82

Table 4.3 Statistics of generated random walk pseudo-corpora ranging from 1m to 3m pseudo-sentences in size. Statistics are presented in groups based on hyperparameters: we first present size, then minimal sentence length, then direction. Rows presenting data on corpora with a 1-word sentence minimum are shaded cyan, 2-word sentence minimum are shaded magenta and 3-word sentence minimum are shaded orange.

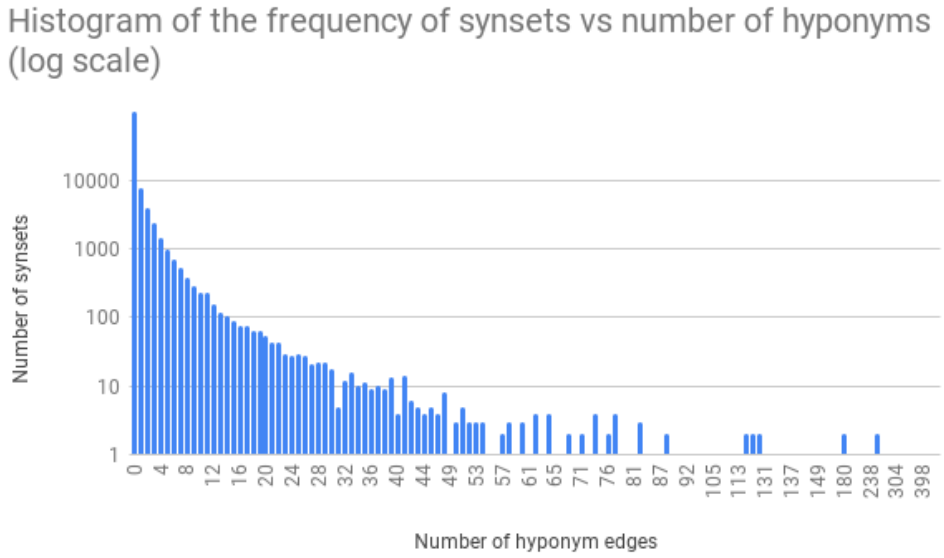
example, the average sentence length for the 500k.both.2w/s is 4.8, and the average sentence length for the 2m.both.2w/s corpus is also 4.8 tokens per sentence. This holds for any other analogous combination, which strongly suggests that there is a common underlying distribution affecting these pseudo-corpora, which is not affected by their size (in terms of pseudo-sentences, i.e. random restarts), but rather by other parameters such as the dampening factor (α), the minimum sentence length and the shape of the graph (i.e. directionality).

Furthermore, the number of tokens also varies largely depending on the latter two hyperparameters. Not surprisingly, we see that in corpora with a higher sentence length minimum the number of tokens is consistently larger than in corpora with a lower sentence length minimum. However, most interestingly, both average sentence length and absolute number of tokens are strongly impacted by the hyperparameter of direction. Regardless of the number of sentences, the corpora generated by only walking up the taxonomy create the longest sentences on average and have the largest number of tokens, while exclusively walking down the taxonomy generates the shortest sentences and the lowest number of tokens, and allowing both directions during the walk creates a sort of middle ground where the corpora are slightly larger than only going down, but much smaller than only going up.

Such behaviour is a direct consequence of the shape of the WordNet taxonomy and the distribution of edges between nodes, as shown in Figure 4.1. The taxonomy is a tree structure with the majority of nodes positioned near the bottom of the tree. Consequently, as there are only a handful of nodes near the top, each time the random walk restarts, it is far more likely to start the random walk at a leaf node somewhere at the bottom of the taxonomy, rather than at the top. Therefore, if the walk is only allowed to go up, on the majority or restarts it will be able to traverse the taxonomy for a comparatively larger number of nodes before either α kicks in, or it reaches the top and has nowhere to go. Conversely, if the walk is constrained to only move down the taxonomy then on most restarts the walk will only be able to take a few steps before it has nowhere to go and is forced to terminate. Finally, the reason that allowing



(a) Hypernym edge distribution



(b) Hyponym edge distribution

Figure 4.1 Distribution of hypernym/hyponym edges between all synsets in WordNet.

both directions in the walk generates shorter sentences than going only up is because almost by definition, a synset can have only 1 hypernym, but several hyponyms. This is seen in Figure 4.1a where most synsets have only one or even zero hypernyms, while larger numbers of hypernyms are much rarer and do not go beyond 5. Contrasting this with Figure 4.1b

which shows that, while most synsets have zero hyponyms, the number of possible hyponyms a synset can have is as high as 398, and there are thousands of nodes that can have up to 20 hyponyms. This means that at a point in the walk where the algorithm is at anode which has both a hyponym and hypernym connection, it is more likely to choose a node that is directed downward for the next step in its walk. In doing so, it behaves more similarly to the algorithm that only goes down and generates shorter sentences than the upward one.

Repeated sentences. Tables 4.2 and 4.3 also present statistics on the amount of repetition in the corpora, in terms of identical sentences. We define identical sentences as two sentences whose bags of words contain the same words (effectively disregarding word order). Given that the vocabulary is limited by what can be found in WordNet, the more we walk the graph, the bigger the chance that the same nodes will be visited, likely via the same paths, and thus identical sentences will be generated. Indeed, the data shows that the more sentences there are in the corpora, the more repeated sentences they have. We hypothesised that this would be beneficial for the eventual taxonomic embeddings, as a certain amount of repetition should reinforce the connections between words, separating information from noise. Our in-depth research on pseudo-corpus sizes has confirmed this hypothesis (Maldonado et al., 2019), but with the caveat that there is a plateau after which growing the size of the random walk pseudo-corpus yields no additional benefits.

However, the number of sentences is not the only factor controlling the amount of repetition in the corpora: the directionality and minimum sentence length hyperparameters also have a strong impact on the percentage of repeated sentences. Regardless of the number of restarts, when looking at corpora with a 3-words per sentence minimum (shaded orange), the highest percentage of repeated sentences appears in corpora generated by walking down the hierarchy, and allowing both directions generates the lowest percentage, whereas corpora generated going up fall somewhere in the middle. Given that the "down" corpora have the shortest sentences, as well as the lowest number of words, it is much more likely for their

sentences to be the same, as any variation between the sentences generally arises from the random restart, rather than the path of the random walk. Meanwhile, corpora that allow both directions have the most options with regards to the path of the random walk, resulting in high sentence variability and a low percentage of repeated sentences.

Interestingly, the above observation regarding repetition in 3-word sentence minimum corpora does not hold consistently for corpora with a 2-word sentence minimum. Walking down does generate the highest percentage of repeated sentences for both the 2w/s and 3w/s hyperparameter. However, in the 1m 2w/s corpora the lowest percentages of repeated sentences are found in corpora generated from only walking up the taxonomy, and it is only in the 2m corpus that lowest percentage comes from both directions being allowed. This switch between 1m and 2m 2w/s corpora in terms of which direction constraint generates the least number of repeated sentences is peculiar, but given how small the differences are, it is likely that there are confounding effects at play here. We suspect that with the 2w/s corpora allowing both directions makes them more similar to the random walk down, which generates a higher number of short sentences that are then repeated. Once the corpus becomes large enough, this effect is then mitigated and the true effect of the variability comes to the fore. Meanwhile, this effect is not present in the 3w/s corpora because eliminating 2-word sentences compensates for that effect.

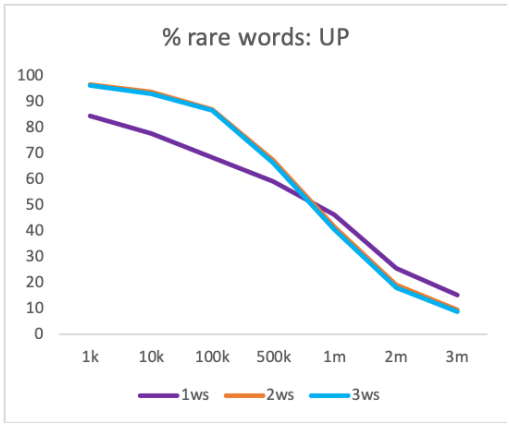
Vocabulary. Tables 4.2 and 4.3 also present statistics on vocabulary size. Naturally, the larger the corpus (both in terms of sentences and tokens), the larger the vocabulary. When comparing the impact of minimal sentence lengths, the vocabulary covered is overall slightly lower in corpora with a 3-word sentence minimum than ones with a 2-word sentence minimum. This difference is small in corpora going up and in both directions, but the difference is quite stark when comparing vocabularies of corpora generated going down (a difference of roughly 8,000-10,000 words). Similarly, when comparing directions, going down produces corpora with the least WordNet coverage, and going in both directions yields

the highest coverage. Again, this is a directly related to the number of tokens and average sentence length. Due to the nature of the random walk going downward the paths are short and there is not much variety, so the vocabulary coverage is significantly lower. Interestingly, allowing for both directions yields a corpus that consistently has almost full coverage, even in the medium-sized corpora, whereas only going up produces a smaller vocabulary in the smaller corpora, but soon catches up as the size increases.

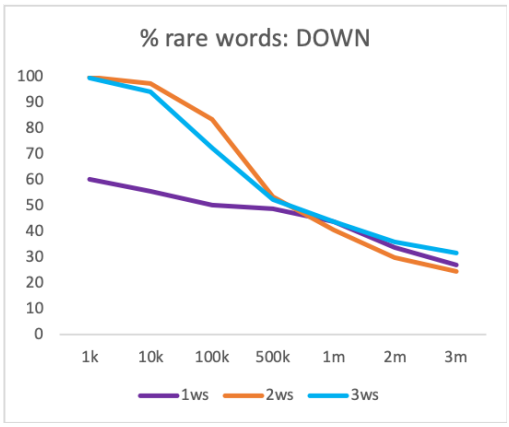
Rare words. Finally, we consider rare words in the generated pseudo-corpora, as previous research has highlighted difficulties in training embeddings for rare words in natural corpora (Lazaridou et al., 2017; Pilehvar and Collier, 2017; Pilehvar et al., 2018; Khodak et al., 2018; Schick and Schütze, 2020) and we suspect they could play an important role in embeddings trained on pseudo-corpora as well. We define a word type as rare if it appears in the pseudo-corpus less than 10 times in a sentence with at least one other word in context[3].

We calculate the percentage of rare words versus the full vocabulary. Values are presented in Tables 4.2 and 4.3, and their plots in Figure 4.2. Overall, the percentage of rare words gets smaller as corpus size increases, as more and more words appear over 10 times. However the hyperparameters seem to have different effects on this value depending on corpus size as well. For the 500k corpora, the highest percentage of rare words are in corpora generated by only going up, while the lowest percentage are in corpora generated when the walk is allowed to proceed in both directions. All percentages are slightly lower for corpora with a 3-word sentence minimum when compared to corpora with a 2-word sentence minimum. The percentage of rare words drops off much quicker for corpora generated by only going

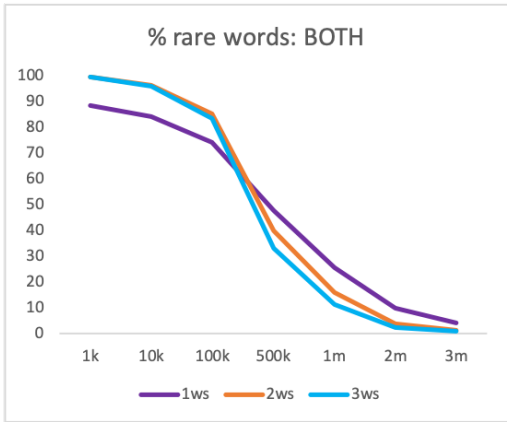
[3]The requirement of at least one other word in context for an instance of a word to be counted towards its rare word frequency extends the standard definition of rare words, which generally just considers word occurrences without considering the context of these occurrences. This extension is necessary with our pseudo-corpora because, unlike natural corpora, 1-word sentences occur quite frequently if the random walk is allowed to traverse a disconnected graph. Instances of words in 1-word sentences should not count towards the word frequencies considered for the definition of rare words for word embedding because these isolated instances provide no contextual information for the word and hence are of no use towards modelling a good taxonomic representation for that word. (Note that for corpora generated with a minimum sentence length hyperparameter > 1 this definition of rare words becomes simply: words which occur less than 10 times in the pseudo-corpus.)



(a) Direction: up



(b) Direction: down



(c) Direction: both

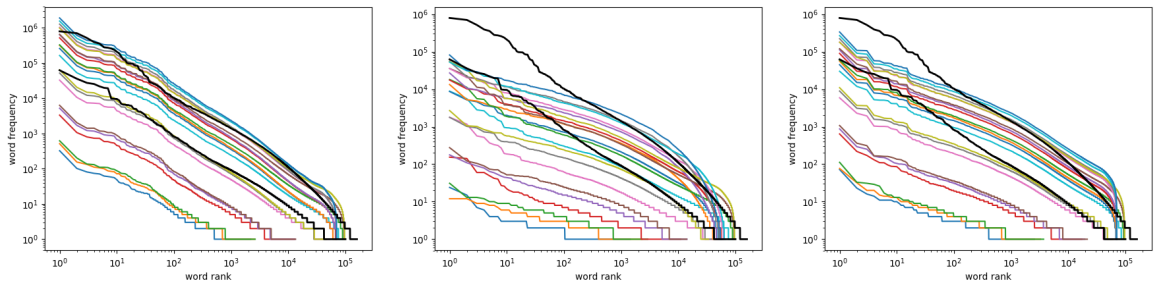
Figure 4.2 Percentage of rare words plotted against the different sizes of pseudo-corpora. Each graph represents corpora generated in one direction (up, down and both respectively) and displays 3 curves for corpora with a 1-, 2- and 3-word sentence minimum (respectively shaded purple, orange and blue)

up compared with corpora generated by only going down. Consequently, even though the up direction generates corpora with the highest percentage of rare words in the smaller sizes, this percentage quickly drops as the corpus size increases. Hence, corpora of 3m sentences generated by only going up have a smaller percentage or rare words compared with the 3m corpora generated by only going down. This is a consequence of the much more drastic increase in number of tokens between the two corpus varieties. The upward corpora consistently have roughly twice as many tokens as the downward corpora of the same number of sentences. Overall, the corpus with the smallest percentage—only 0.82% of rare words in the vocabulary—is the one generated with 3m sentences, a 3 word-sentence minimum and allowing the walk to move in both directions. Likely, this is because it is generated from the graph with the most connections, and hence an overall higher coverage; at the size of 3 million sentences, it would have traversed most of the taxonomy several times over, thereby significantly reducing the number of rare words.

These are all properties that arise as a consequence of these corpora being artificially generated. They are all stem from the graph structure of the WordNet taxonomy and from the way the random walk algorithm has traversed this graph. However, we also looked at word distributions and noticed interesting trends that seem to indicate similarities with natural corpora, so we investigate this further.

4.4 Scaling Linguistic Laws of Natural Languages

Regularities in the frequency of text constituents have been summarized in the form of *linguistic laws* (Gerlach and Altmann, 2014; Altmann and Gerlach, 2016). Linguistic laws provide insights on the mechanisms of text production, which can, in a limited sense, also be understood as a proxy for language or thought production.



(a) Direction: up (b) Direction: down (c) Direction: both

Figure 4.3 Zipf distributions of two natural corpora (shaded black) and all our pseudo-corpora grouped according to the direction parameter.

4.4.1 Zipf's Law

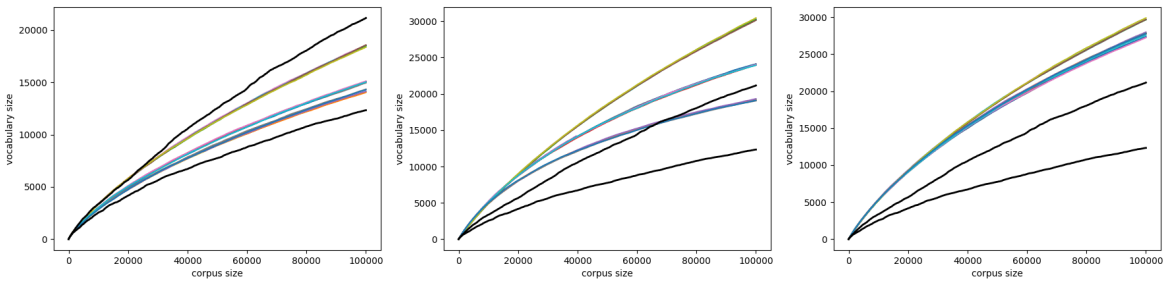
One of the best known linguistic laws is *Zipf's Law* (Zipf, 1949). It states that the frequency F of the r^{th} most frequent word (i.e. the fraction of times it occurs in a corpus) scales as

$$F_r \propto r^{-\lambda}, \forall\, r \gg 1 \tag{4.1}$$

Zipf's Law is approximated by a Zipfian distribution which is related to discrete power law probability distributions. Here, λ is the scaling exponent and it has been found to be ≈ 1.0 for natural languages. In other words, in a natural language corpus, the frequencies of words are inversely proportional to their ranks in the frequency table, i.e. the most frequent word will occur about twice as often as the second most frequent word, three times as often as the third most frequent word, etc.

4.4.2 Heaps' Law

Heaps' Law is another linguistic law, also a scaling property of language, which describes how vocabulary grows with text size. Consider n be the length of a text and $v(n)$ be its vocabulary size. Then Heaps' law is formulated as follows:



(a) Direction: up (b) Direction: down (c) Direction: both

Figure 4.4 Heaps' law of two natural corpora (shaded black) and all our pseudo-corpora grouped according to the direction parameter.

$$v(n) \propto n^{\beta}, \forall\, n \gg 1 \tag{4.2}$$

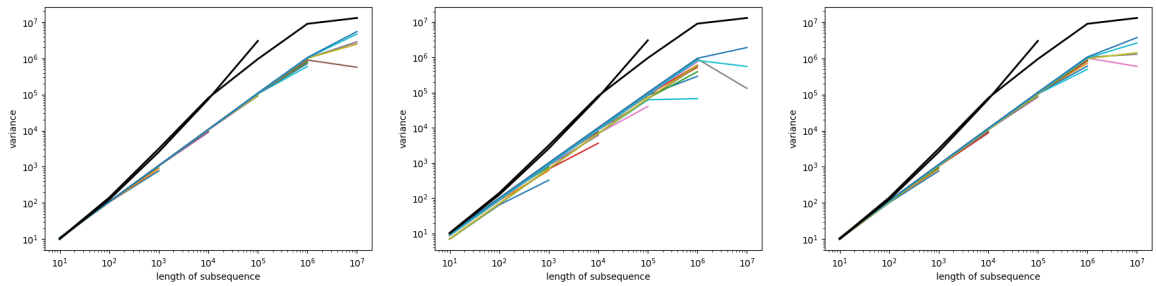
where the exponent for the Heaps' law for natural languages is found to be $0 < \beta < 1$. In other words, Heaps' law means that as more instances of natural text are gathered, there will be diminishing returns in terms of discovery of the full vocabulary from which the distinct terms are drawn, i.e. as the text gets bigger, there will be less and less new additions to the vocabulary[4].

4.4.3 Ebeling's Law

We also consider *Ebeling's Law*, which studies the growth of variance of individual components (e.g. letters or words in text) in relation to the subsequence length l. Described by Takahashi and Tanaka-Ishii (2019), for a set of words W, let $y(k,l)$ be the number of occurrences of word $w_k \in W$ for all subsequences of length l of the original dataset. Then,

$$m(l) = \sum_{k=1}^{|W|} m_2(k,l) \propto l^{\eta} \tag{4.3}$$

[4]In natural language the vocabulary is theoretically infinite, so gathering more text should never reach 100% coverage, however the vocabulary in WordNet is finite and will eventually reach a saturation point, given enough repeated random walks. Still, we consider the possibility that the distributions might be similar before reaching the finite vocabulary limit.



(a) Direction: up (b) Direction: down (c) Direction: both

Figure 4.5 Ebeling's law of two natural corpora (shaded black) and all our pseudo-corpora grouped according to the direction parameter.

$m_2(k,l)$ is the variance of $y(k,l)$. Here, $m(l)$ relates to l with a power-law relationship with exponent η. Ebeling and Pöschel (1994) showed that the Bible has $\eta = 1.69$. In other words, there is a specific relationship between the size of a sequence of natural text and the variance of words that occur in that sequence. It can be understood as describing the variety of words found in a text, which becomes higher as the subsequence size increases.

Taking these natural linguistic laws into account, we test whether our pseudo-corpora uphold such laws, so as to investigate their own naturalness. We have compared the Brown corpus (Francis, 1964) and a relatively small chunk of wikitext-2 (Merity et al., 2016) with all our generated pseudo-corpora. Figures 4.3, 4.4 and 4.5 display the plots of Zipf's, Heaps' and Ebeling's laws respectively for the two natural corpora (shaded black) as well as all our generated pseudo-corpora. In addition to plotting the individual curves, we employed *Kolmogrov-Smirnov (KS) Distance* to compare the pseudo-corpora against the natural corpora. The Kolmogorov-Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. In our case, we check *KS* distance between the natural and pseudo-corpora for Zipf's, Heap's and Ebeling's law.

Our analysis revealed that the KS distance between our 2 natural corpora is consistent with the distance between the natural and synthetic corpora, indicating consistent variations

for Zipf's, Heaps' and Ebeling's law. For both our natural and synthetic corpora, $\lambda \approx 1.1$ and $\beta \approx 0.9$. In this case, it is fair to assume that our pseudo-corpora maintain these properties of natural language. This finding is important because it indicates that word representations derived from taxonomic pseudo-corpora would have similar limitations to representations derived from natural text. For example, previous research has shown that learning good embeddings for rare words in natural corpora can be a challenge (Lazaridou et al., 2017; Pilehvar and Collier, 2017; Pilehvar et al., 2018; Khodak et al., 2018; Schick and Schütze, 2020). We explore the impact of rare words in the pseudo-corpora on embedding performance in Section 4.5.

Though our test of KS distance confirms that all the pseudo-corpora follow certain natural distributions, it is still interesting to note the slight variations in the generated plots. Uniformly, the 'up' pseudo-corpora most closely match the natural corpora, the 'down' pseudo-corpora do so to a much lesser degree, while 'both' fall somewhere in the middle. This indicates that the directionality hyperparameter also enables us to simulate slightly different underlying graph structures, accounting for the variation in the statistical distributions. These figures reinforce the fact that the nature of the random walk algorithm, the structure of the graph and the paths that are walked do have an impact on the resulting pseudo-corpus. They might not impact the fact that they reflect scaling laws found in natural language, but they still have an impact on the distributions of the words in the generated text, which can propagate down the line if integrated into various machine learning and language modelling pipelines.

4.5 Training, validation and analysis

After generating all the corpora, we train word embeddings and validate them by evaluating their performance on word similarity benchmarks. To validate that training on our pseudo-corpora can generate taxonomic embeddings, and for the purpose of methodological

consistency with Goikoetxea et al. (2015) and comparability with their work, in this section we only evaluate and discuss embeddings obtained from the word2vec SGNS encoder. However, in our probing experiments in Chapter 5 where we compare taxonomic and thematic embeddings, we also train taxonomic embeddings using the GLoVe encoder for a more comprehensive comparison.

Additionally, as this evaluation serves more as validation of the taxonomic embeddings, we do not perform it for all the generated pseudo-corpora described in Section 4.3, but only for a subset of them. Specifically, we evaluate embeddings trained on pseudo-corpora between 500 thousand (500k) and 2 million (2m) pseudo-sentences, and we do not evaluate embeddings trained on pseudo-corpora that contain 1-word pseudo-sentences. These choices are motivated by the findings of our sister-experiments (Maldonado et al., 2019), which have already extensively evaluated corpora with 1-word sentences, and have shown that the amount of taxonomic information begins to saturate between 500k and 2m sentences. We also exclude 1-word-sentence corpora from the evaluation because we wish to be more strict in our definition of taxonomic embeddings, restricting it to only words with taxonomic connections in WordNet.

4.5.1 Training word2vec taxonomic embeddings

We trained our taxonomic embeddings using the 2017 version of Pytorch SGNS, a publicly available off-the-shelf implementation[5] of the skip-gram with negative sampling (SGNS) algorithm, introduced by Mikolov et al. (2013a). We only made minor data-handling optimisations, but the objective function is not modified in any way.

The vectors were computed with SGNS using a window of five words on both sides of a sliding focus word, without crossing sentence boundaries. Twenty words were randomly selected from the vocabulary based on their frequency as part of the negative sampling step

[5] `https://github.com/theeluwin/pytorch-sgns`

of the training. The frequencies in this weighting were smoothed by raising them to the power of $\frac{3}{4}$ before dividing by the total. All vectors produced by the SGNS system have 300 dimensions and trained for 30 epochs. We trained separate embeddings on each combination of the three hyperparameters and report evaluations of the best performing epoch.

4.5.2 Validation

We evaluate the performance of our embeddings on five different benchmarks:

- **SimLex-999** (Hill et al., 2015). It consists of 999 word pairs whose similarity judgements emphasise taxonomic and synonymic similarity over all other semantic relations, which receive very low similarity scores. Semantic similarity systems tend to perform much worse on SimLex-999 than on mixed thematic-taxonomic benchmarks such as WordSim-353 and SemEval-17.

- **WordSim-353** (Finkelstein et al., 2002)[6]. It consists of 353 word pairs and is an older and more established semantic similarity dataset that conflates thematic and taxonomic similarities.

- **SemEval-17** (Camacho-Collados et al., 2017). The English[7] test set from the SemEval 2017 Task 2 challenge. It consists of a set of 500 pairs of words, multiword expressions and entities from a wide range of domains. These 500 pairs are uniformly distributed across a scale of five degrees of similarity that range from total dissimilarity to complete synonymy, with thematic and taxonomic similarities falling at different points along this scale. Notably, thematic similarity is considered to be at a lower scale than taxonomic similarity.

- **Princeton evocation** dataset (Boyd-Graber et al., 2006). It consists of 13,176 word pairs which have been human-annotated and assigned a value of "evocation" repre-

[6] `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`
[7] Though other languages are available, we only focus on the monolingual similarity task in English.

senting how much the first concept brings to mind the second. Though this is not really a word similarity task as it does not directly test for either thematic or taxonomic knowledge, it can be approached with the same methodology, so we include it in our evaluation as a sanity check, fully expecting our embeddings to underperform on it.

- **WordNet-paths**. We suspect none of the above benchmarks are ideally suited to evaluating our taxonomic embeddings, as they are all based on human judgements on a sometimes broad idea of word similarity, which often confounds taxonomic and thematic relations (Kacmajor and Kelleher, 2019), yet we are modelling taxonomic information specifically. For this reason, in addition to the above benchmarks, we develop another test set, inspired by the work of Pedersen et al. (2004):[8] we take the word pairs from SimLex, and replace the human similarity judgements with a WordNet similarity measure (based on the distances in the graph). We refer to this benchmark as WordNet-paths. It serves as another sense check and an appropriate test set for our taxonomic embedding model, ensuring that the evaluations are comparing like for like.

As is common practice, we evaluate our model by computing a Spearman correlation score between the cosine similarity of the word vectors from our model and the scores in our benchmarks (be it human judgement or WordNet distance). Table 4.4 presents the results alongside the percentage of rare words in a given benchmark.

4.5.3 Results

The aim of this experiment is not to beat state of the art scores on these benchmarks, but rather to investigate different WordNet taxonomic structures generated by the random walk hyperparameters and their impact on rare words and performance of word embeddings trained on the pseudo-corpora. Our main hypothesis is that the direction constraint of the random

[8] `http://wn-similarity.sourceforge.net`

	simlex		ws353		semeval		evoc		wn-paths	
corpus	**%rare**	**score**	**%rare**	**score**	**%rare**	**score**	**%rare**	**score**	**%rare**	**score**
500k-up-2w/s	2.63	39.03	8.01	39.24	11.81	37.23	5.26	7.93	2.63	52.89
500k-down-2w/s	2.53	19.22	6.86	21.23	10.47	20.46	3.72	4.46	2.53	41.86
500k-both-2w/s	1.14	32.56	2.97	42.76	4.83	38.12	1.31	9.87	1.14	56.31
500k-up-3w/s	2.92	37.07	7.09	34.65	11.60	35.70	4.71	8.61	2.92	50.60
500k-down-3w/s	2.97	31.26	8.70	33.34	10.06	27.51	5.26	4.13	2.97	49.12
500k-both-3w/s	1.04	34.84	2.75	45.53	4.72	40.36	1.10	10.61	1.04	57.00
1m-up-2w/s	1.24	41.73	3.20	43.34	5.85	39.56	2.08	8.61	1.24	53.44
1m-down-2w/s	1.09	30.46	3.43	41.69	6.26	35.09	2.08	6.90	1.09	47.56
1m-both-2w/s	0.50	40.55	0.92	48.25	1.75	40.93	0.44	11.14	0.50	57.60
1m-up-3w/s	1.19	42.28	2.75	39.75	5.85	40.51	2.19	9.75	1.19	54.15
1m-down-3w/s	1.93	36.37	5.03	42.65	8.11	36.19	4.05	5.48	1.93	51.15
1m-both-3w/s	0.35	42.13	0.69	46.59	1.33	39.16	0.33	10.93	0.35	57.73
2m-up-2w/s	0.59	42.58	1.14	44.38	2.77	39.61	0.77	8.63	0.59	53.52
2m-down-2w/s	0.69	34.87	1.14	41.79	4.00	36.75	0.99	5.62	0.69	47.67
2m-both-2w/s	0.15	43.28	0.46	47.03	0.41	40.48	0.22	10.95	0.15	58.00
2m-up-3w/s	0.50	43.40	1.14	43.97	2.46	39.71	0.77	9.65	0.50	54.01
2m-down-3w/s	1.04	36.80	3.43	44.29	5.44	35.17	2.41	4.85	1.04	49.47
2m-both-3w/s	0.05	43.28	0.46	47.51	0.31	40.35	0.22	11.14	0.05	56.55

Table 4.4 Results for all embeddings trained on various corpora, showing Spearman correlation scores for best epoch per corpus trained on, as well as the percentage of rare words in a given benchmark. Cells shaded green represent the lowest percentage of rare words and the highest Spearman score obtained in the given group of embeddings on a given benchmark. Cells shaded red represent the highest percentage of rare words and the lowest Spearman score on the given group.

walk has an effect on the percentage of rare words in the resulting corpus, which in turn affect the performance of the trained embeddings.

With that in mind, we examine Table 4.4. Our results interpretation examines models in groups of three: for each benchmark we compare correlation scores and percentage of rare words between corpora of the same size and minimum sentence length, but different direction constraints (up, down or both). Cells shaded green represent the lowest percentage of rare words and the highest Spearman score obtained in the given group of embeddings on a given benchmark. Cells shaded red represent the highest percentage of rare words and the lowest Spearman score on the given group.

Our highest correlation scores come from the WordNet-paths benchmark, which is not surprising as this dataset reflects most closely what our models have learned: taxonomic relations in WordNet. The highest overall score comes from the largest corpus, but looking at the different groups of different-sized corpora, the best performing model is always the one allowing both directions in the random walk, which generates the lowest percentage of rare words. Our hypothesis is clearly confirmed on this benchmark, where all the best scores come from corpora with the lowest percentage of rare words, while the lowest scores come from corpora with the highest percentage of rare words in two out of six cases.

In contrast with WordNet-paths, our worst performance is achieved on the evocation benchmark. This is also expected, as the evocation benchmark models a relationship between words that is very different in nature from the purely taxonomic relationship that we model here. This, together with the fact that our best correlation scores come from the WordNet paths benchmark, supports the evidence that our embeddings do indeed contain a taxonomic representation of words. Yet in spite of the correlation scores being so low, our hypothesis holds here as well: in each group of comparable embeddings, the highest score comes from pseudo-corpora that traversed both directions, and generated the fewest rare words. The

lowest scores stem from corpora with the highest percentage of rare words in five out of six cases.

As expected, we achieve much higher correlation scores on the remaining three benchmarks. Though the highest scores are achieved on WS-353, the overall performances between benchmarks are comparable insofar as they all model word similarity and relatedness. Our hypothesis holds just as consistently when examining the results on SemEval-17 and WS-353, where five out of six times and six out of six times respectively, the best performing model stems from a corpus that yields the lowest percentage of rare words, while the inverse holds four out of six times.

SimLex-999 seems to be somewhat of an outlier among these benchmarks. This is peculiar because, though it is more similarity-focused, the nature of the relations should not be that different from the one in WS-353 and SemEval-17. Our hypothesis still holds in the larger corpora (2m-2w/s, 2m-3w/s and 1m-3w/s), but in the smaller ones the lowest percentage of rare words is produced by the corpora allowing both directions, yet the highest scores actually come from the corpora produced going up. Given that the inconsistencies happen in the smaller corpora, it is possible that this is just an unlucky sample, or that the interplay of confounding factors has a stronger effect in the smaller corpora and negatively affects the performance of the corpora allowing both directions.

Overall, the distribution of best-worst models is fairly consistent across the 5 benchmarks. The best models are those going in both directions, and 2-word sentence minimum models are usually slightly outperformed by 3-word sentence models, though the differences are marginal. Unsurprisingly, models trained on corpora allowing both directions also consistently produce the lowest percentage of rare words, and more often than not these models have the best scores.

4.6 Resource publication

Goikoetxea et al. provide an implementation of their pseudo-corpus generation algorithm[9]. However, due to the significant differences our algorithm has introduced, as outlined in Section 4.2, and the the special use cases required for our research which focused on analysing how the shape of knowledge graph affects the properties of the synthesized corpora, we reimplemented the algorithm using NLTK's Python version of WordNet (Bird and Loper, 2004)[10]. We have also made our random walk code publicly available via GitHub[11], and have included a detailed guide on how to use the provided scripts. In addition to a script for generating pseudo-corpora with varying hyperparameters, there is also a script for calculating basic corpus statistics, and a script for calculating a word similarity score using word embeddings and cosine similarity.

As far as our corpora, we have published all resources related to our research on Arrow@TUDublin[12], which is Technological University Dublin's official archive and data repository. This includes an archive of all 81 pseudo-corpora that were generated for our research (Klubička et al., 2020). They are published in the form of a compressed archive of text files, and once extracted each individual pseudo-corpus can be used with our statistics script, or as training corpora for any word embedding system[13].

Additionally, we have also used the data repository as an archive for our taxonomic word embeddings, which we trained on the above pseudo-corpora (with some exceptions). This includes a total of 72 pretrained taxonomic word embedding models that were trained for the purposes of our research (Maldonado et al., 2019; Klubička et al., 2019)[14].

[9] http://ixa2.si.ehu.eus/ukb/
[10] http://www.nltk.org
[11] https://github.com/GreenParachute/wordnet-randomwalk-python
[12] https://arrow.dit.ie
[13] https://arrow.dit.ie/datas/9/
[14] https://arrow.dit.ie/datas/12/

4.7 Conclusion

In this chapter we have expanded our understanding of the random walk algorithm using the WordNet taxonomy as a case study. We examined the relationship between the structure of the underlying knowledge graph, the properties of the pseudo-corpora generated from the graph, and the performance of the embeddings trained on these pseduo-corpora. We found that the pseudo-corpora derived from WordNet's taxonomy are not as artificial as one might expect, as they resemble natural corpora at a statistical level. We attribute these properties to the underlying tree structure of the graph from which the pseudo-corpora are built. We also train word embeddings on these corpora to study the impact of these properties on the embedding performance on word similarity evaluation tasks. Our evaluations confirm a successful modelling of taxonomic relations, and on most benchmarks our data supports the hypothesis that the ratio of rare words in a pseudo-corpus affects embedding performance.

Understanding the properties of the pseudo-corpora generated from a knowledge graph structure can inform how the random walk should be designed and run for any graph. For example, knowing that a tree-like graph structure results in pseudo-corpora exhibiting Zipfian properties is useful as it highlights the presence of rare words in the corpora. As the vocabulary of the lexical resource is finite, the problem of rare words within the generated pseudo-corpora can be addressed by ensuring that the pseudo-corpus is large enough so that even the relatively rare words appear frequently enough to learn adequate embeddings. This perspective helps in answering questions such as: *how large should a pseudo-corpus be* and *which combination of hyperparameters will provide the best taxonomic embeddings for a taxonomic probing task?*

Though this might seem obvious, an important takeaway is that the properties of any pseudo-corpus generated from a knowledge graph will be affected by the properties of that graph: its structure and node connectivity will be reflected by the word distributions in the generated corpora, thus impacting the resulting embeddings. We do not claim that any graph

structure will exhibit the exact properties we found, but rather that this kind of analysis should be considered when using a random walk algorithm.

Taking a step back, we acknowledge a possible limitation of this work, which ties into a more general consideration about any vector space model: Karlgren and Kanerva (2021) argue that while local subspaces in a semantic space are well-defined and can represent commonalities between words located within, the global structures of the vector space are arbitrary and any meaningful relationship that might be ascribed to the distance between words in subspaces that are far apart are only spurious. From this follows that for any given word or subspace in a semantic space there is a *horizon of interest* beyond which drawing connections to other words does not allow for any salient inference of meaning or relatedness. This consideration is even more pertinent when it comes to our taxonomic embeddings, which only reflect the hypernymy relationship and are built on a sparsely connected graph. This likely gives them a limited ability to model relationships between words that are far apart from each other in the taxonomy. Certainly, the presupposition is that, to a certain degree, the embedding models are able to encode the distance (i.e. the number of edges) between words that have no immediate taxonomic relationship but are connected via other nodes, and we see some evidence of this in our results. However, due to the nature of the random walk algorithm, the pseudo-sentences often end up being short, even with a high sentence-length limit, so the most accurate word representations will likely reflect contexts of words that are closely linked, rather than words which are taxonomically far apart. Hence, evaluating these embeddings on the task of word similarity, as captured by word similarity benchmarks and measured via cosine similarity, might not be the best tool for measuring the taxonomic knowledge encoded in them. Especially when some of the word pairs in these benchmarks are so far apart that they may as well belong to separate taxonomies, e.g. *brainstorming* and *telescope*, or *elementary school* and *forest*. It is not entirely fair to examine a notion of relatedness between these word pairs using embeddings that mainly

encode immediate hyponym-hypernym relationships; in such a scenario, the cosine similarity measure is arguably not an adequate indicator of the nature of their taxonomic relationship, when there is barely one to speak of.

To obtain more direct assessment of whether these embeddings encode the relationships they were trained on—hypernym-hyponym relations—we need to evaluate them on a more appropriate task using a more suitable evaluation framework. We thus develop a hypernym-hyponym probing task and apply our probing with noise method to our favoured taxonomic embeddings in Chapter 5, in order to examine how well they encode direct taxonomic information compared to thematic embeddings, and to explore where in the embeddings this information is contained.

Chapter 5

Probing Taxonomic vs Thematic Embeddings

Having trained word embeddings on WordNet random walk pseudo-corpora as described in Chapter 4, our evaluations indicate that these embeddings encode taxonomic information, and allow us to make some relative inferences on which pseudo-corpora yield embeddings that are better at encoding such information, in turn allowing us to make an informed decision on which embeddings are best suited for a taxonomic probing task. In this chapter we examine their behaviour on a probing task more suitable than word similarity: given that our taxonomic embeddings most explicitly encode a hypernym-hyponym relationship between words, we design a hypernym-hyponym classification task and apply our *probing with noise* method (as described in Chapter 3) to perform an intrinsic, relative evaluation. In order to draw broader comparisons, we apply the same evaluation framework to our taxonomic SGNS embeddings and to pretrained thematic SGNS embeddings. To confirm whether the findings will hold on a different encoder, we run the same set of experiments on GLoVe embeddings. Narratively, the experiments described in this chapter serve as a simple, focused example of the application of our probing with noise method and illustrate the types of insight it can provide, before moving on to a larger suite of experiments in Chapters 6 and 7.

5.1 Hypernym-Hyponym Prediction

While hypernym detection is not the focus of the thesis, we still present an overview of some notable work on this topic in order to establish a context and connect our work with the wider literature.

Hypernymy, understood as the capability to relate generic terms or classes to their specific instances, lies at the core of human cognition and plays a central role in reasoning and understanding natural language (Wellman and Gelman, 1992). Two words have a hypernymic relation if one of the words belongs to a taxonomic class that is more general than that of the other word. For example, the word *vehicle* belongs to a more general taxonomic class than *car* does, as *car* is a type of *vehicle*. Hypernymy can be seen as an *IS-A* relationship, and more practically, hypernymic relations determine lexical entailment (Geffet and Dagan, 2005) and form the *IS-A* backbone of almost every ontology, semantic network and taxonomy (Yu et al., 2015). Given this, it is not surprising that modelling and identifying hypernymic relations has been pursued in NLP for over two decades (Shwartz et al., 2016), and successfully doing so has proven useful in downstream tasks and applications such as question answering (Prager et al., 2008; Yahya et al., 2013), textual entailment and semantic search (Hoffart et al., 2014; Roller et al., 2014; Roller and Erk, 2016), web retrieval, website navigation and records management (Bordea et al., 2015).

That being said, while research on hypernym detection has been plentiful, work that applies any probing framework to identify taxonomic information in embeddings is scarce, and the existing work does nor probe for it directly, but rather infers taxonomic knowledge from examining higher-level tasks. For example, Ettinger (2020) identified taxonomic knowledge in BERT, but rather than probing BERT embeddings using a probing classifier, BERT's masked-LM component was used instead and its performance was examined on a range of cloze tasks, where the goal was to fill an incomplete sentence with the missing word.

One of the relevant findings was that BERT can robustly retrieve noun hypernyms in this setting, demonstrating that BERT is very strong at associating nouns with their hypernyms.

Ravichander et al. (2020) build on Ettinger's work and investigate whether probing studies shed light on BERT's systematic knowledge, and as a case study examine hypernymy information. They devise additional cloze tasks to test for consistency in predictions, and demonstrate that BERT often fails to consistently make the same prediction in slightly different contexts. They conclude that BERT's ability to correctly retrieve hypernyms in cloze tasks is not a reflection of larger systematic knowledge, but possibly an indicator of lexical memorisation (Levy et al., 2015).

Aside from this recent focus on BERT, not much other work has been done in the space of probing embeddings for taxonomic information, or specifically hypernymy probing. However, work on modelling hypernymy has a long history that stretches back before BERT and pretrained language models.

Traditionally, identifying hypernymic relations from text corpora has been addressed with two main approaches: *pattern-based* and *distributional* (Wang et al., 2017). Pattern-based methods exploit the co-occurrence of a hyponym and its hypernym in a textual corpus (Hearst, 1992; Navigli and Velardi, 2010; Boella and Di Caro, 2013; Flati et al., 2014, 2016; Gupta et al., 2016; Pavlick and Paşca, 2017). Earlier work was mostly unsupervised and leveraged various interpretations of the distributional hypothesis. One such interpretation is the concept of distributional generality (Weeds et al., 2004; Clarke, 2009), based on the observations that more general words tend to occur in a larger variety of contexts than more specific words. For example, it should be possible to replace any occurrence of *cat* with *animal* and so all of the contexts of *cat* must be plausible contexts for *animal*. However, not all of the contexts of *animal* would be plausible for *cat*, e.g., "the monstrous animal barked at the intruder". Lenci and Benotto (2012) took this notion further and hypothesised that more general terms should have high recall and low precision, which would thus make it

possible to distinguish them from other related terms such as synonyms and co-hyponyms. Based on this reasoning, they developed a variant of the distributional generality measure that allowed them to identify hypernyms. Other measures for identifying hypernyms have also been developed: for example, SLQS (Santus et al., 2014) is an entropy-based measure based on the hypothesis that the most typical linguistic contexts of a hypernym are less informative than the most typical linguistic contexts of its hyponyms.

Conversely, distributional approaches rely on a distributed representation for each observed word, capable of identifying hypernymic relations between concepts even when they do not co-occur explicitly in text. Some distributional approaches leverage similarities between vectors to model a hypernymy relationship. As briefly discussed in Section 4.1, cosine measures on word embeddings pairs give an indication of the overall *semantic relatedness* of the word pairs they represent (Turney and Pantel, 2010), without specifying the type(s) of semantic relation(s) the two words hold. There have been endeavours to modify the similarity function or train word embeddings that emphasise one semantic relation over another in order to facilitate better hypernymy models. For example, Rei and Briscoe (2013) experimented on parser lexicalisation and found that a WeightedCosine directional similarity measure performs well on the task of detecting hypernyms. In a similar vein, Nguyen et al. (2017) developed the Hypervec algorithm by adapting the skip-gram objective function to emphasise the asymmetric hypernym-hyponym relations. In essence they convert the similarity function into a hypernym-relation function, resulting in a cosine similarity measure that does not reflect word "similarity", but rather that one word is the hypernym of the other.

However, most distributional hypernymy models have been supervised, mainly based on using word embeddings as input for classification or prediction (Baroni et al., 2012; Santus et al., 2014; Fu et al., 2014; Espinosa-Anke et al., 2016; Ivan Sanchez Carmona and Riedel, 2017; Nguyen et al., 2017; Pinter and Eisenstein, 2018; Bernier-Colborne and Barrière, 2018; Nickel and Kiela, 2018; Cho et al., 2020; Mansar et al., 2021).

Interestingly, Roller et al. (2018) studied the performance of both pattern-based and distributional approaches on several hypernymy tasks and found that simple pattern-based methods consistently outperform distributional methods on common benchmark datasets, showing that pattern-based models provide important contextual constraints which are not captured in distributional methods. Finally, Shwartz et al. (2016) have shown that pattern-based and distributional evidence can be effectively combined within a neural architecture to improve prediction results.

We highlight the work of Weeds et al. (2014), who also used a supervised approach and demonstrated that it is possible to predict whether or not there is a specific semantic relation between two words given their distributional vectors. Their work is especially relevant to ours as it shows that the nature of the relationship one is trying to establish between words informs the operation one should perform on their associated vectors: e.g. using the difference between the vectors for pairs of words is appropriate for an entailment task, whereas adding the vectors works well for a co-hyponym task. This is a consideration we need to take into account in the construction of our hypernym-hyponym probing task.

In terms of evaluation benchmarks for modeling hypernymy, they have generally been designed such that in most cases they are reduced to binary classification (Baroni and Lenci, 2011; Snow et al., 2005; Boleda et al., 2017; Vyas and Carpuat, 2017), where a system has to decide whether a hypernymic relation holds between a given candidate pair of terms. Criticisms to this experimental setting point out that supervised systems tend to benefit from the inherent modeling of the datasets in the hypernym detection task, leading to lexical memorization phenomena (Levy et al., 2015; Santus et al., 2016; Shwartz et al., 2017). In this respect, there has been work attempting to alleviate this issue by including a graded scale for evaluating the degree of hypernymy on a given pair (Vulić et al., 2017).

In an alternative approach to the problem, Espinosa-Anke et al. (2016) proposed to frame it as Hypernym Discovery: rather than a binary classification of the relationship, given the

search space of a domain's vocabulary, and given an input term, discover the term's best (list of) candidate hypernym(s). This addressed one of the main drawbacks of the earlier evaluation criterion and inspired Camacho-Collados et al. (2018) to construct a full-fledged hypernym discovery benchmark covering multiple languages and knowledge domains. The dataset was released as a shared task in *SemEval-2018 Task 9: Hypernym Discovery*, with the goal of expanding the research in hypernymy modelling.

Indeed, in some of our earlier work we participated in this shared task. We trained thematic SGNS embeddings on in-domain corpora and used a standard cosine similiarity calculation to output hypernym candidates (Maldonado and Klubička, 2018), which made for a competitive unsupervised system. However, we do not report on this work in the thesis beyond its mention here in related work, due to it falling out of the scope of the thesis: we did not employ probing and did not use taxonomic embeddings to solve the shared task. Still, having engaged with hypernyms in the past has informed some of the research directions and the design of the task and dataset presented in this chapter.

While we acknowledge the hypernym discovery task as introduced by Camacho-Collados et al. (2018) as an important hypernymy benchmark, and the cloze tasks used by Ettinger (2020) as an enlightening probing scenario, we suspect neither is suitable for our probing experiments, for which we require a simpler task that is better at teasing out the hypernym-hyponym relationship we wish to probe for. Specifically, rather than an open-ended hypernym discovery task, or even a binary relationship prediction task, we opt to construct a more direct taxonomic task: predicting which word in a pair is the hypernym, and which is the hyponym. This approach is informed by the work of Weeds et al. (2014), as our setup implicitly takes into account the asymmetric nature of the hypernym-hyponym relationship.

5.2 Hypernym-Hyponym Probing Task Dataset Creation

As stated by Conneau et al. (2018), a probing task needs to ask a simple, non-ambiguous question, in order to minimise interpretability problems and confounding factors. Hence, for our experiments we needed a probing task that does not just use hypernym-hyponym taxonomic knowledge to solve an unrelated or semi-related classification task, but rather a task that probes for taxonomic knowledge directly. To this end, we constructed a dataset that is derived from WordNet (Fellbaum, 1998), comprised of all of its hypernym-hyponym pairs. That way each word pair shares only an immediate hypernym-hyponym relationship between the candidate words: a word in a pair can be either a hyponym or a hypernym of the other, there is no other option. This dataset contains a total of 329,396 hypernym-hyponym pairs.

However, in our experiments we wish to apply our method to both taxonomic and thematic encoders. Given that the vocabulary coverage of our taxonomic embeddings is constrained to WordNet, and the probing task dataset was also derived directly from WordNet, the pretrained thematic embeddings which were trained on natural corpora may not have the same coverage, which would give our taxonomic embeddings and advantage. Additionally, the pretrained GloVe and SGNS embeddings also have different coverage between them, as they were not trained on the same corpora. We wish to mitigate confounders as much as possible by comparing like for like, so to retain a high integrity of interpretation when comparing models, we opted to filter down the dataset and only evaluate on the intersection of vocabularies of the four models—we only include word pairs that have a representation for both words in all four embedding models. This step reduced the dataset size to 246,747 word pairs.

Note here that one of the goals of our work is to use our probing with noise method to learn about embeddings and the way they encode different types of information in vector space. We assert that a prediction of the relationship between a pair of words cannot be fairly done without the classifier having access to representations for both words in the pair. Yet, our probe is a classifier which can only take a single vector as input (see Section 5.3).

Informed by the work of Weeds et al. (2014) we considered options such as averaging or summing the individual word vectors, but found that these were not suitable to our framework as they muddled the notion that the classifier is receiving two separate words as input. We instead opted to concatenate the word vectors in question and pass a single concatenated vector to the classifier (similar to approaches used by Adi et al. (2017)). Though even in this scenario the classifier has no explicit indication that it is receiving a representation of a pair of words as input, if there is a signal in the individual word vectors that differentiates the hypernyms from the hyponyms, and the probe is powerful enough, then it should be able to pick up on it. This approach allows us to formulate the task as a positional classification task: given a pair of words, is the first one the hypernym or the hyponym of the other? We can then assign each instance in the corpus a binary label—0 or 1—representing the class of the first word in the pair. The probe can then predict if the left half of the vector is the hyponym (0) of the right half, or whether it is its hypernym (1).

Finally, given the imbalance in the distributions of hypernyms and hyponyms in WordNet (see Section 4.3), a smaller number of words will be hypernyms, while a larger number will be hyponyms. We want to avoid the probe memorising the subset of words more likely to be hypernyms, but rather to learn from information encoded in the (differences between) vectors themselves. In an attempt to achieve this, we balance out the ratio of class labels by duplicating the dataset and swapping the hypernym-hyponym positions and labels. Before duplicating, we also define a hold-out test set of 25,000 instances, so as to exclude the possibility of the same word pair appearing in both the train and test split—thus, the probe will be evaluated only on unseen instances. This duplication resulted in a final dataset of 493,494 instances, of which 50,000 comprise the test set and 443,494 comprise the training set. Here are some example instances from the dataset:

- *0, north, direction*
- *1, direction, north*

- *0, hurt, upset*
- *1, upset, hurt*

5.3 Experimental Design

Having established a dataset, we can test the proposed method of *probing with noise*, as described in Chapter 3, and compare the evaluations of taxonomic and thematic embeddings, as well as different encoders.

5.3.1 Embedding Models

In our experiments we compare our taxonomic SGNS embeddings to pretrained thematic SGNS embeddings, as well as make an analogous comparison of newly trained taxonomic GloVe embeddings and pretrained thematic GloVe embeddings.

word2vec (SGNS) For taxonomic SGNS representations we use the embeddings described in Chapter 4. We opt for embeddings trained on the pseudo-corpus that yielded the highest Spearman correlation score on the wn-paths benchmark (see Section 4.5), i.e. the corpus with 2 million sentences, with the walk going both ways and with a 2-word minimum sentence length. The lack of a directionality constraint provides higher vocabulary coverage and a smaller proportion of rare words, while the 2-word minimum sentence length limit ensures that we only have representations for words that are part of WordNet's taxonomic graph and have at least one hypernym-hyponym relationship, which makes them suitable for this task.

For the thematic embeddings we use a pretrained SGNS model, and opt for the gensim[1] word2vec implementation which was trained on a part of the Google News dataset (about 100 billion tokens) and contains 300-dimensional vectors for 3 million words and phrases[2].

[1] `https://radimrehurek.com/gensim/`
[2] word2vec-google-news-300

GloVe To train taxonomic GloVe embeddings, we use a popular Python implementation of the GloVe algorithm[3] and apply it to the same 2m-both-2w/s pseudo-corpus to obtain taxonomic embeddings, using the same approach as described in Section 4.2[4].

For the thematic GloVe embeddings we use the original Stanford pretrained GloVe embeddings[5], opting for the larger common crawl model, which was trained on 840 billion tokens and contains 300-dimensional embeddings for a total of 2.2 million words.

Note that when we concatenate the two word embeddings required for an instance in the train or test set, they become a 600-dimensional vector which is then passed on as input to the probe.

5.3.2 Probing Classifier and Evaluation Metric

In all our probing experiments (Chapters 5, 6 and 7), the embeddings are used as input to a Multi-Layered Perceptron (MLP) classifier, which predicts their class labels. We used the scikit-learn MLP implementation (Pedregosa et al., 2011) using the default parameters[6].

The choice of evaluation metric used to evaluate our probes is not trivial, as we want to make sure that it is reliably reflecting a signal captured in the embeddings, especially in an imbalanced dataset where the probe could learn the label distribution, rather than detect a true signal related to the probed phenomenon. As some of the datasets that we use in our experiments do have an imbalanced distribution (e.g. the hypernym-hyponym dataset in Chapter 5 or the idiomatic usage dataset in Chapter 6), it is crucial to select a suitable performance metric.

[3] `https://github.com/maciejkula/glove-python`

[4] We used the following training parameters: window=10, no_components=300, learning_rate=0.05, epochs=30, no_threads=2. Any other parameters are left as default.

[5] `https://nlp.stanford.edu/projects/glove/`

[6] activation='relu', solver='adam', max_iter=200, hidden_layer_sizes=100, learning_rate_init=0.001, batch_size=min(200,n_samples), early_stopping=False, weight init. $W \sim \mathcal{N}\left(0, \sqrt{6/(fan_{in} + fan_{out})}\right)$ (scikit relu default). See: `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`

Due to its intuitiveness, accuracy would be anyone's first port of call, but it is not suited for imbalanced datasets: a model could report high accuracy by blindly labelling every sample as positive or negative if the imbalance was too high. This could be accounted for by establishing all-yes or all-no performance baselines, but there are more appropriate evaluation metrics to use in such cases. The F1 score and Area Under Precision Recall curve (AUC-PR) are both suitable for the standard imbalanced scenario where the positive class is in the minority, as both focus on the identification of positive samples. However, in our experiments on idiomatic usage (see Chapter 6), the positive class (idiomatic usage) is actually the majority class, which makes metrics like F1 and AUC-PR less than ideal. Meanwhile, metrics like AUC-ROC (Area Under Receiver Operating Characteristic Curve) and Matthews correlation coefficient (MCC) reflect the classifier's performance on both positive and negative classes and are also suitable for imbalanced datasets. Furthermore, an empirical comparative study by Halimu et al. (2019) showed that both AUC-ROC and MCC are statistically consistent with each other, however, AUC-ROC is more discriminating than MCC. Therefore we selected the AUC-ROC score[7] as the metric for our probe evaluations. We use it consistently throughout our cohort of experiments, even in cases where the label distributions are balanced, in order to facilitate consistency and comparability between datasets and results.

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR, also known as sensitivity or recall) against the false positive rate (FPR) at various threshold settings. When using normalised units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html`

randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). Hence, the AUC-ROC metric varies between 0 and 1, with an uninformative classifier yielding 0.5.

Finally, as mentioned in Section 3.5, to address the degrees of randomness in the method, we train and evaluate each model 50 times and report the average score of all the runs, essentially bootstrapping over the random seeds (Wendlandt et al., 2018). Additionally, we calculate a confidence interval to make sure that the reported averages were not obtained by chance, and report it alongside the results.

5.3.3 Chosen Noise Models

As described in Section 3.3.2, we remove information from the norm by sampling random norm values and scaling the vector dimensions to the new norm. However, considering that vectors have more than one calculable norm, the scaling can be done to match more than one norm value. While we have examined the effects of scaling to both the L1 and L2 norms, which are most widely used in NLP, in order to streamline the results presentation, henceforth when discussing norm ablations we only report results pertaining to scaling to the L2 norm. Specifically, we sample the L2 norms uniformly from a range between the minimum and maximum L2 norm values of the respective embeddings in our dataset[8].

To ablate information encoded in the dimension container, we randomly sample dimension values and then scale them to match the original norm of the vector (see Section 3.3.1). Specifically, we sample the random dimension values uniformly from a range between the minimum and maximum dimension values of the respective embeddings in our dataset[9]. We

[8]Thematic SGNS: [0.6854, 9.3121]
Taxonomic SGNS: [2.1666, 7.6483]
Thematic GloVe: [3.1519, 13.1196]
Taxonomic GloVe: [0.0167, 6.3104]

[9]Thematic SGNS: [-1.5547, 1.7109]
Taxonomic SGNS: [-1.8811, 1.7843]
Thematic GloVe: [-4.2095, 4.0692]
Taxonomic GloVe: [-1.3875, 1.3931]

expect this to fully remove all interpretable information encoded in the dimension values, making the norm the only information container available to the probe.

Applying both noise functions together on the same vector should remove any information encoded in it. In this case, the probe should have no signal in the actual embeddings to learn from, which would be akin to training it on random vectors.

Finally, we use the vanilla SGNS and GloVe word embeddings in their respective evaluations as vanilla baselines against which all of the introduced noise models are compared. Here, the probe has access to both information containers—dimension and norm—as well as class distributions from the training set. However, it is also important to establish the vanilla baseline's performance against the random baselines: we need to confirm that the relevant information is indeed encoded somewhere in the embeddings.

5.4 Experimental Results

Detailed experimental evaluation results for taxonomic and thematic embeddings on the hypernym-hyponym probing task are presented in Tables 5.1 and 5.2. Note that all cells shaded light grey belong to the same distribution as random baselines on a given task, as there is no statistically significant difference between the different scores; cells shaded dark grey belong to the same distribution as the vanilla baseline on a given task; and all cells that are not shaded contain a significantly different score than both the random and vanilla baselines, indicating that they belong to different distributions.

5.4.1 SGNS

Starting with the results of the pretrained, thematic SGNS embeddings (THEM), Table 5.1 shows that the random baselines perform comparably to each other, as would be expected, and their score indicates no ability to discriminate between the two classes. We can see that

SGNS				
Model	THEM		TAX	
	auc	±CI	auc	±CI
rand. pred.	.5000	.0009	.4997	.0009
rand. vec.	.5001	.0012	.5001	.0011
vanilla	.9163	.0004	.9256	.0003
abl. N	.9057	.0004	.9067	.0005
abl. D	.5039	.0008	.5294	.0010
abl. D+N	.4998	.0010	.5002	.0009

Table 5.1 Experimental results on word2vec SGNS models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

the vanilla representations significantly outperform the random baselines, indicating that at least some taxonomic information is encoded in the embeddings. Having established the vanilla results as a baseline for the ablations, we can examine which information container encodes the relevant information: dimension or norm.

The norm ablation scenario causes a statistically significant drop in performance when compared to the vanilla baseline. In principle, this indicates that some information has been lost. If instead of the norm, we ablate the dimension container, we see a much more dramatic performance drop compared to vanilla, indicating that much more information has been removed. Unsurprisingly, the probe's performance in the scenario where we apply both noising functions drops to $\approx$0.5, and the difference in its performance when compared to random baselines is not statistically significant, so there is no pertinent information left in these representations.

Notably, once just the dimension container is ablated from these vectors, its performance drops to extremely low levels and approaches random baseline performance, yet it does not quite reach it—as small as it is, the difference is statistically significant, indicating that not all information has been removed. Arguably, given how minor this difference is, while

significant, it is not a very convincing argument in favour of the norm's role in encoding taxonomic information.

However, we detect a stronger signal when examining our taxonomic SGNS embeddings (TAX). Yet again, the random baselines perform comparably, while the vanilla baseline significantly outperforms them. Not only that, but it also significantly outperforms the THEM vanilla baseline, confirming that our WordNet random walk taxonomic embeddings encode more taxonomic information than thematic embeddings.

In terms of the container ablations, we observe similar behaviour as in the THEM example: the norm ablation scenario causes a statistically significant drop in performance when compared to the vanilla baseline; ablating the dimension container yields a larger performance drop compared to vanilla, but does not quite reach the random-like performance achieved when ablating both containers.

Here the difference in scores between ablating just the dimensions and ablating both dimensions and norm is also significantly different from random, but notably also an order of magnitude larger than in the THEM example. This indicates that our taxonomic SGNS embeddings use the norm to encode taxonomic information more so than the pretrained thematic embeddings. To confirm this finding, we examine the behaviour of GloVe embeddings in the analogous experiments.

5.4.2 GloVe

First looking at the pretrained, thematic GloVe embeddings (THEM) in Table 5.2, we see yet again that the random baselines behave as expected. The vanilla GloVe performance dramatically outperforms the baselines, but the scores drop when the norm is ablated. After ablating the dimension container, there is a substantial drop in the probe's performance and it is immediately comparable to random baselines with no statistically significant difference. Furthermore, performance does not significantly change after also ablating the norm.

GloVe				
Model	THEM		TAX	
	auc	±CI	auc	±CI
rand. pred.	.4999	.0011	.4998	.0010
rand. vec.	.5001	.0010	.5001	.0008
vanilla	.9327	.0004	.8824	.0005
abl. N	.9110	.0004	.8435	.0008
abl. D	.5002	.0008	.6621	.0008
abl. D+N	.5000	.0011	.5006	.0011

Table 5.2 Experimental results on GloVe models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

Meanwhile, the taxonomic GloVe embeddings tell a different story. Firstly, while the vanilla embeddings outperform the random baselines, they perform much worse than THEM vanilla GloVe, indicating an inferior representation for the hypernym-hyponym prediction task, even though they were trained on WordNet random walk pseudo-corpora (we discuss this further in Section 5.6). Ablating the norm causes a significant drop in performance, but it is nowhere near the random performance reached when ablating both dimensions and norm. This is a really strong signal that indicates the norm is at least partially responsible for encoding some hypernym-hyponym information. This also confirms the same finding in SGNS, demonstrating that our taxonomic embeddings use the norm to encode taxonomic information more so than pretrained thematic embeddings.

5.5 Post Hoc Experiment: Dimension Deletions

One of the expectations which guided our experimental design was that providing the probe with a concatenated vector of two word embeddings would allow it to infer the asymmetric relationship between the two candidate words and use that as a signal to make its prediction. To ensure this, we have taken some steps to mitigate lexical memorisation (see Section 5.2).

We would also expect memorisation to already be hampered by the nature of the probing task itself, given that, aside from the root and leaf nodes, many words in the taxonomy take on the role of both hypernym and hyponym. In other words, it is never the case that e.g. *dog* is always a hyponym or always a hypernym—the word can take on either role across different candidate pairs in the dataset.

Still, there is a concern that the models could have memorised other regularities encoded in the individual word representations and used that information to make predictions. For example, while many candidate words can indeed be both hyponyms or hypernyms, given the tree structure of the taxonomy and the distribution of edges (see Figure 4.1), the frequencies at which a word takes on a hypernym or hyponym role are still skewed. It is thus more likely that any given word will be a hyponym than a hypernym, and it is possible that the embeddings implicitly encode the frequency at which a word takes on a hypernym role, versus a hyponym role.

To account for this confounding factor and to measure its impact, we run an additional batch of probing experiments to establish another set of baselines that help compare against this confounder, which is specific to this particular probing task. In staying consistent with the ablational nature of the *probing with noise* method, in this post hoc batch of experiments we examine the impact of two scenarios on the probe's performance: a) what if the probe's input was only one word, and b) what if the probe's input was only half of each word vector in the pair?

We denote this line of enquiry as *post hoc deletion experiments*, given that in practice a) can be considered as deleting half of the concatenated vector, and b) as deleting one half each vector before concatenating. The crucial difference between the two scenarios is that in a) the probe can only learn from the one word vector without having any access to a representation of the other word, meaning it cannot infer a relationship between the two candidate words and can only predict whether the candidate word is a hyponym or a

hypernym by relying on the probability derived from its frequency. Conversely, in b) the probe is given a representation for both vectors, meaning if there is a relationship between them it could be leveraged, however the individual vectors are truncated, meaning that half of the dimension information is lost from both words, making the representations inferior to the vanilla setting[10].

We ran these experiments for both the taxnonomic and thematic SGNS and GloVe embeddings and when performing deletions assessed the impact of both halves of the vectors. All dimension deletion results are included in Tables 5.3 and 5.4, where scenario a) is denoted as *del.ct.1h/2h* (deleted 1st/2nd half of concatenated vector) and scenario b) is denoted as *del.ea.1h/2h* (deleted 1st/2nd half of each vector). When comparing the deletions of the different halves, in cases where there is a statistically significant difference between their scores, the lower of the two scores is marked with an asterisk. Examining the results provides some relevant insights.

5.5.1 SGNS

Unsurprisingly, deleting half of the vector in either scenario causes a statistically significant drop in performance when compared to vanilla. We also observe a larger drop in both del.ct. settings versus the del.ea. settings, which confirms that predicting a word's relationship to an "imaginary" other word is the more difficult task.

However, strikingly, the performance is also significantly above random, which indicates that the probe likely did learn some frequency distributions from the graph, as it has nothing else to learn from. It is possible that this is a reflection of the inherent imbalance in the dataset, as there is a large number of leaf nodes in the taxonomic graph, which can only be hyponyms.

[10]This choice is motivated by a desire to make this setting comparable to a) in terms of dimensionality—had we simply compared it to vanilla, it would have the advantage of having access to twice as many dimensions.

SGNS				
Model	THEM		TAX	
	auc	±CI	auc	±CI
rand. pred.	.5000	.0009	.4997	.0009
rand. vec.	.5001	.0012	.5001	.0011
vanilla	.9163	.0004	.9256	.0003
del. ea. 1h	.8929	.0004	.8998*	.0005
del. ea. 2h	.8927	.0004	.9039	.0004
del. ct. 1h	.8496	.0004	.8525	.0004
del. ct. 2h	.8495	.0004	.8523	.0003

Table 5.3 Experimental results on SGNS deletions models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

Even still, the significant difference in scores between the two settings demonstrates that having access to both words, even at the cost of half the information in each word's dimensions, is more informative than having a full representation of a single word, *indicating that the probe is inferring the relevant relationship between them.*

Additionally, it is worth noting that the performance is not always comparable between each respective vector half: in the case of TAX del.ea.1h/2h, though small, the difference in scores between the two halves is statistically significant, whereas this is not the case in the three remaining settings where there are no significant differencese between deleting the 1st half of the vector, versus the 2nd half.

5.5.2 GloVe

In terms of deletions, the GloVe results echo the findings on SGNS in most settings. Deleting half of the vector in either scenario causes a significant performance drop, which is largely above random performance, and the drop is larger in the *del.ct.* setting versus the *del.ea.* setting, providing further indication that, while there is an inherent imbalance in the underlying

GloVe				
Model	**THEM**		**TAX**	
	auc	±CI	auc	±CI
rand. pred.	.4999	.0011	.4998	.0010
rand. vec.	.5001	.0010	.5001	.0008
vanilla	.9327	.0004	.8824	.0005
del. ea. 1h	.9120*	.0003	.8727	.0005
del. ea. 2h	.9179	.0004	.8730	.0006
del. ct. 1h	.8522	.0004	.8405	.0004
del. ct. 2h	.8522	.0004	.8406	.0004

Table 5.4 Experimental results on GloVe deletions models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

data, the probe is inferring the relevant relationship between the candidate words when given a concatenation of two word vectors.

Similar to SGNS, the performance is not always comparable between each respective vector half, however in the case of GloVe it is the THEM *del.ea.* where the difference in scores between the two halves is statistically significant. That said, in both SGNS and GloVe this difference is very small.

5.6 Discussion

There are a number of points to take away from the experimental results presented in this chapter. Firstly, and most importantly for this thesis, they provide strong evidence that *embedding models can use the norm to encode taxonomic information*.

Note, however, that while ablating just the norm causes a drop in performance, we are conscious that this also happens fairly consistently in all of our experiments involving SGNS and GloVe embeddings. As described in our fictional example in Section 3.7, we are wary of taking this result on its own as a strong indicator that the norm itself encodes some of

the task-relevant information. It seems that this drop is relatively small regardless of task or encoder ($<$0.1 in most cases[11]), so it is likely to be an artefact of a particular interaction between information encoded in the dimensions and the norm, or one between the noising function and the embeddings[12], rather than a reflection of the norm encoding task-specific, in this case taxonomic, information.

While we believe norm ablation results on their own should not be considered conclusive evidence of the norm encoding taxonomic information, the remaining scenarios can be considered as a sequence of related ablations and as such can offer more reliable indications. The dimension ablation scenario in tandem with the dimensions and norm ablation scenario provides the relevant insight. Notably:

i) in cases where just the dimension container is ablated from the vectors and its performance drops to above-random, this indicates that the taxonomic information is not contained *only* in the dimension container; ii) furthermore, when the dimension and norm ablation functions are then applied together, which induces a further performance drop comparable to random baselines, this can be taken as evidence that the vectors with ablated dimension information still contain residual information relevant to the task, which is removed when also ablating the norm. We provided an example of this in Section 3.7, but it is important to reiterate the result here: when both i) and ii) hold, this strongly suggests that the norm contains some of the relevant information *regardless of what is encoded in the vector dimensions*.

We observe a strong example of this in the case of the taxonomic GloVe embeddings, where the AUC-ROC score after ablating the dimension information is still as high as $\approx$0.66, meaning that the difference of 0.16 points is solely due to the information in the norm. We consider this a very large difference given our understanding of the underlying mechanics,

[11] See also Table 7.1 for additional examples.

[12] Perhaps, given the relatively low dimensionality of the SGNS and GloVe vectors, the introduction of random noise in the norm container disrupts even dimension information sufficiently to cause this slight drop in performance, even though the norm itself does not carry much relevant information.

where it is well known that dimensions typically contain most, if not all information relevant for a task—as an inverse example, in thematic GloVe embeddings, no discernible task-specific information is found in the vector norm, but rather all the information is contained in the dimensions.

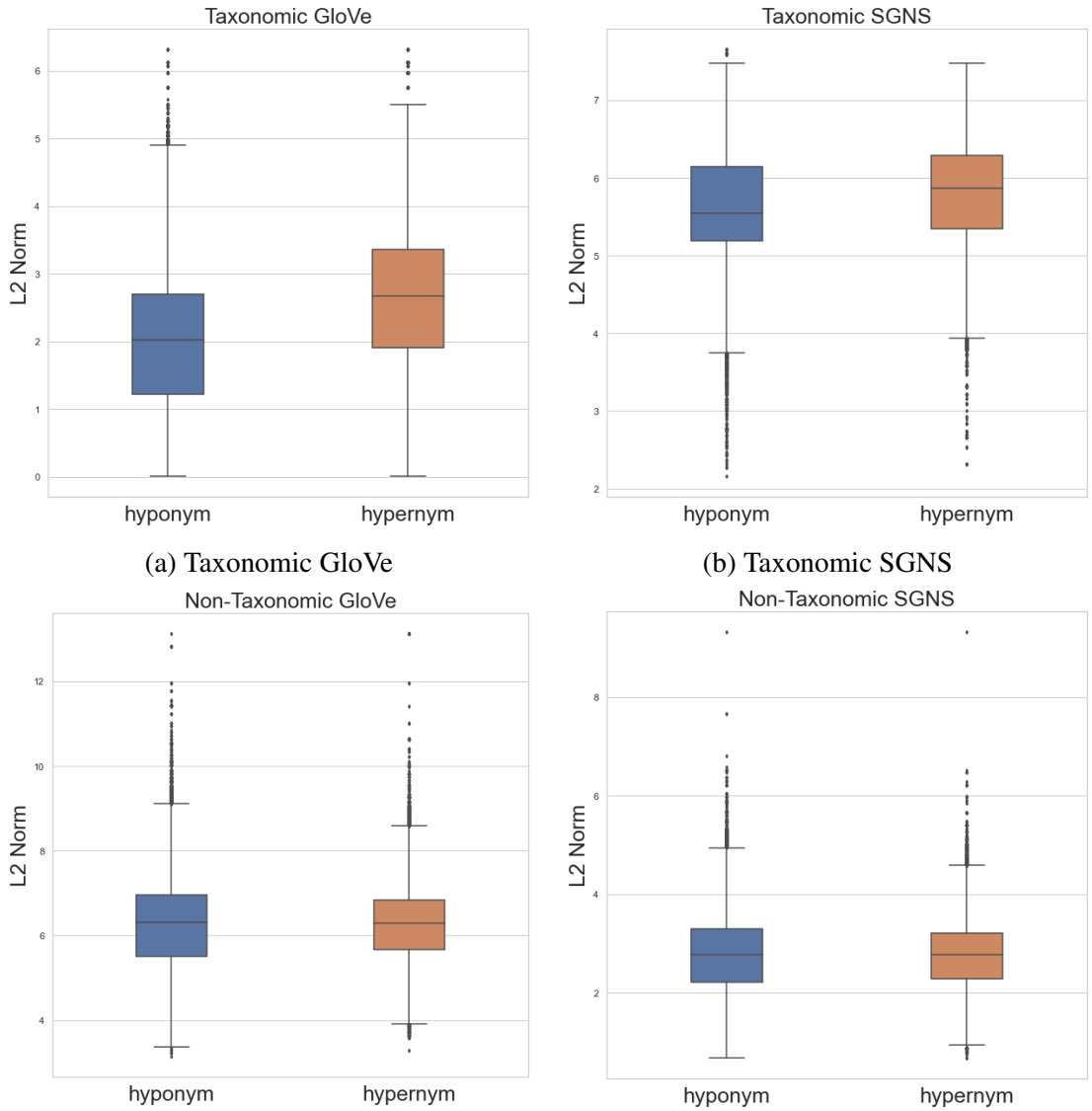
It is also worth noting that in taxonomic GloVe embeddings, ablating the norm causes the most significant drop in performance, much larger than in any analogous scenarios (dropping from ≈0.88 to ≈0.84). In fact, this is the only case in our experiments where we found that deleting half of each word vector before training yields a significantly higher score (≈0.87) than ablating the norm (≈0.84). In tandem, these findings suggest that more information is lost when the norm is ablated than when half of the dimensions are removed. This is a strong indicator that in this case the *norm encodes information that is not at all available in the dimensions*. Certainly, the majority of the information in an embedding is and will always be encoded in the dimensions, but it is striking how much of it is present in the norm in this case.

Generally, when it comes to dimension deletion experiments, it is expected that the performance would drop dramatically in comparison to vanilla embeddings. However, an important takeaway is that in all settings the drop is much smaller than might be expected, being quite close to vanilla performance and largely above random performance. This points to a redundancy within the dimensions themselves, seeing as either half of the vector seems to carry more than half the information required to model the task, indicating that not many dimensions are needed to encode specific linguistic features. This is consistent with the findings of Durrani et al. (2020), who analysed individual neurons in pretrained language models and found that small subsets of neurons are sufficient to predict certain linguistic tasks. Our deletion results certainly corroborate these findings, given how small the drop in the probe's performance is when half the vector is deleted.

Another finding concerns the scores being lower in the setting where half the concatenated embedding is deleted, or rather, when the probe is predicting based on only one word vector. This demonstrates that the probe benefits significantly from having access to a representation of both words, or even just two halves of each word representation, even when it is not explicitly told that it is actually getting two inputs. This indicates that giving the probe access to both allows it to extrapolate a relation between them, which informs the probing classifier's decisions. It is able to pick up on the fact that there is a difference between them which can be helpful in deciding on a label. In the case of our taxonomic embeddings, this difference may very well be the difference in their norms.

To confirm this finding, we investigate the norm differences and find that this interpretation is supported by the actual values of the vector norms in our dataset. We calculate the norms of the individual hypernym and hyponym word vectors in our dataset and present the results in Figure 5.1. Calculating the median norm shows that the difference between hypernym and hyponym norms seems to be minor in both thematic embeddings (GloVe: 6.26 and 6.24; SGNS: 2.78 and 2.76), whereas the difference is an order of magnitude larger in both taxonomic representations (GloVe: 2.03 and 2.67; SGNS: 5.64 and 5.80). The difference is also quite large between taxonomic GloVe and SGNS, and it seems to be what is reflected in our experimental results, which show that GloVe stores the most hypernym-hyponym information in the norm.

These measurements also align with the interpretation that the depth of the taxonomic tree would be mapped to the vector's distance from the origin of the space. Surprisingly, however, it is the opposite of what we would expect. Based on the fact that more frequent words tend to be positioned closer to the origin (Goldberg, 2017), one intuition would be that words positioned higher up in the taxonomy, i.e. words belonging to root nodes, might be positioned closer to the origin of the space, as according to the notion of distributional generality (Weeds et al., 2004) they might be more frequently used in language. On the flip



(a) Taxonomic GloVe

(b) Taxonomic SGNS

(c) Thematic GloVe

(d) Thematic SGNS

Figure 5.1 Box plots depicting the median values of the L2 norm in the different sets of word vectors, split by whether the word is a hyponym or hypernym. There is a marked difference observed between hyponym and hypernym norms in taxonomic GloVe and SGNS, but not in thematic.

side, words much deeper in the taxonomy, e.g. words belonging to leaf nodes, have far fewer connections and appear in much more specific contexts, which makes them far less frequent in natural language. Hence we would expect them to be positioned further away form the origin, in order to denote this separation and specificity of context.

However, the median norm measurements show that, on average, the norm of hypernyms is larger than the norm of hyponyms. This means that hypernyms, which are higher up in the tree, are positioned further away from the origin than hyponyms, which are positioned lower in the tree, and are closer to the origin. Additionally, this is only true for our taxonomic embeddings, but not the thematic ones, where the median norm values are comparable, with no statistically significant differences.

We suspect that the unintuitiveness of this measurement, which appears only in the taxonomic embeddings, is related to the fact that the taxonomic tree is, in a sense, "bottom-heavy". While according to the notion of distributional generality, hypernyms might occur more frequently than hyponyms in natural language, when it comes to a taxonomy, due to the distributions of hypernym-hyponym edges in the graph, the most frequent words are likely positioned at the lower-middle end, and as such are quite numerous (recall Figure 4.1). It is possible that due to the fact that these bottom-adjacent nodes can act as both hypernyms and hyponyms, they invert the seemingly intuitive relationship between frequency and norm. Given that the hypernyms positioned at the very top of the tree would be less frequently traversed by the random walk (which is more likely to go downhill than uphill), they would thus appear less frequently in the pseudo-corpus, and as such seem to end up further away from the origin. This reasoning could also explain the many outliers visible in Figure 5.1. Still, the indication that in taxonomic embeddings *there is a mapping between the taxonomic hierarchy and distance from the origin* is an important finding that warrants more examination.

Admittedly, we are somewhat puzzled by the unintuitiveness of the measurement. Finally, having confirmed that our method is able to successfully identify the separate information containers, we abstract away from the methodological specifics and turn the discussion to differences between the different embeddings—both in terms of architecture and taxonomic/thematic information.

First, we see that the vanilla thematic embeddings, both SGNS and GloVe, encode taxonomic information, and the GloVe model significantly outperforms the SGNS model. This is at least partially due to the fact that the pretrained SGNS and GloVe embeddings were trained on unrelated corpora (Google News vs common crawl respectively), which differ both in terms of size, topic and coverage. The word representations derived from them are likely very different: the corpus that GloVe was trained on is over 8 times larger than the one used to train the SGNS model, and belongs to a different, much more varied genre of text data. It is possible that due to the broader scope and much larger size, these representations reflect more taxonomic knowledge.

Further, these encoders exhibit the opposite behaviour when trained on the same WordNet random walk pseudo-corpus. The scores of the vanilla taxonomic SGNS scores improve upon its thematic version, which is to be expected. Yet the vanilla taxonomic GloVe scores significantly underperform compared to thematic, and are in fact the worst-performing vanilla model in this set of experiments. We would expect it to mirror what was observed in the SGNS experiments and have the taxonomic GloVe outperform the thematic one. Given that both taxonomic SGNS and GloVe were trained on the same random walk corpus, it is possible that this difference in behaviour is due to an interaction between the model architecture and the training data, and we speculate that a range of factors could be at play.

As described in Section 2.2.2, GloVe is trained on a global word-word co-occurrence matrix within a context window, whereas SGNS is trained by predicting the context based on an input word. While neither model's context window crosses sentence boundaries when training embeddings, it is still possible that there is an interaction between certain properties of the pseudo-corpora and the way the embeddings are generated. We suspect that the boundaries between contexts are more strict in the taxonomic corpora than in natural corpora. The generated pseudo-sentences are quite short (5.64 tokens on average) compared to natural sentences, and there is only the small local context to learn from.

Meanwhile, GloVe being a global model uses aggregate co-occurrences from the whole corpus for each occurrence of a target word within a context window. As the target window is set to 10, it is longer than most pseudo-sentences in the corpus and thus in reality takes the full sentence into account as a target word's context. As such it is designed to benefit from being trained on a much larger and more diverse resource. It is likely that the short sentences and limited vocabulary in our pseudo-corpora make GloVe's word-word co-occurrence matrix relatively sparse: most words in the corpus only co-occur with a very small number of other, similarly frequent words in the taxonomy.

In contrast, SGNS only ever takes individual instances of local context into account when generating embeddings, which is precisely what our taxonomic pseudo-corpus offers. We expect that this makes the pseudo-corpus a resource better suited to the architecture of SGNS as it lends itself to its approach of extracting meaningful relationships between words.

All that being said, while the above factors could be influencing this behaviour, we suspect that the answer is much simpler: the dominant factor is training corpus size. The random walk pseudo-corpus used for training taxonomic embeddings was only about 9 million tokens in size, whereas SGNS's training data had 100 billion tokens, and GloVe's had 840 billion. Hence it is not surprising that a GloVe model trained on a small and relatively sparse pseudo-corpus underperforms when compared to one trained on a 840-billion-token natural corpus. If anything, it is encouraging that training an SGNS model on a 9-million-token pseudo-corpus improves vanilla performance scores over one trained on a 100-billion-token natural corpus.

Overall, in spite of the fact that the worst-performing vanilla model is taxonomic GloVe, it is important to highlight that out of the 4 types of embeddings, taxonomic GloVe also encodes the most taxonomic information in the norm. We base our interpretation of this result on the following: i) in many embeddings there is a high correlation between the norm and word frequency (Goldberg, 2017), and ii) WordNet pseudo-corpora reflect hypernym-hyponym

frequencies and co-occurrences. We suspect the principal signal that plays a role in the way taxonomic embeddings encode taxonomic knowledge is precisely these word co-occurrences, which GloVe is designed to capture. In turn, the norm can be seen as analogous to the hierarchical nature of taxonomic relationships and becomes the most accessible place to store this information. The thematic corpora reflect thematic co-occurrences and frequencies and hence GloVe does not store taxonomic information in the norm, as such relations are not hierarchical in nature. We suspect that thematic embeddings will store other types of linguistic information in the norm, and explore this in Chapters 6 and 7.

5.7 Conclusion

In this chapter we tested our hypothesis that the norm can be a carrier of certain types of information. To answer this question, we applied our *probing with noise* method to two different types of word representations—taxonomic and thematic—each generated by two different embedding algorithms—SGNS and GloVe—on a newly-designed taxonomic probing task of hypernym-hyponym classification.

The most relevant findings for the overall thesis are that (a) the norm is indeed a separate information container, (b) the norm can carry some information pertinent to the hypernym-hyponym probing task, (c) different encoders utilise the norm to varying degrees, (d) the norm container can sometimes be "empty", (e) the majority, but not all, of the task-relevant information is encoded in the dimensions, and (f) while in some cases there can be redundancy between the information encoded in the norm and dimensions, other times the norm can encode information that is not at all available in the dimensions. Jointly, all these findings validate our *probing with noise* method as a viable approach in identifying where in an embedding certain information is encoded.

In addition, our results show that all the tested embeddings, even thematic ones, contain taxonomic information, as they can be used to predict the task well, and we have found

evidence that the probe is, at least to some degree, using the relationship between the candidate words as a predictive feature, even in spite of possible lexical memorisation. We also show that in the case of SGNS, taxonomic embeddings outperform thematic ones on the task, demonstrating the usefulness of taxonomic pseudo-corpora in encoding taxonomic information. Indeed, our method has shined a light on the importance of the norm, showing that the taxonomic embeddings use the norm to supplement their encoding of taxonomic information. In other words, random walk corpora can improve taxonomic information in representations, which is not the case for natural corpora.

But even thematic embeddings trained on natural corpora still encode taxonomic information in the dimensions quite well, especially in the case of GloVe, even though this was not its explicit training goal. However, the fact that it does not use the norm to do so raises the question of whether its norm encodes some type of thematic information instead. Naturally, we would like to know what other kinds of insights our method can provide beyond just a hypernym-hyponym probing task.

Having exhausted the insights obtainable in taxonomic embeddings and taxonomic information, and intrigued by the high performance of GloVe embeddings on the taxonomic task, we are motivated to explore the other end of the semantic spectrum and investigate more broadly the many types of non-taxonomic information that might be encoded by thematic embeddings. We explore this research direction in the following chapters.

Chapter 6

Probing Static vs Contextual Embeddings: Idiomatic Usage

In the previous chapters we have explored taxonomic embeddings in detail and have shown that even thematic GloVe embeddings are good at encoding taxonomic information. We have also shown that the GloVe model has the capability of encoding information in the norm, as seen in taxonomic GloVe embeddings. However, its thematic version does not do this, raising the question of whether there is perhaps some non-taxonomic information that thematic GloVe does use the norm for. This line of reasoning motivates us to move away from taxonomic information and to investigate non-taxonomic probing tasks in order to identify what other kinds of linguistic information might be encoded in the norm.

Since we are now shifting the focus towards thematic representations, we cannot omit contextual encoders such as BERT from our study, given their current prominence. BERT also captures thematic information, but is more advanced than GloVe and is able to generate different, contextualised representations for each word. Context is important for non-taxonomic and thematic relations and so a contextual encoder like BERT is an obvious choice for the application of our method. Hence, in addition to GloVe, we run the same sets of experiments on the transformer-based BERT. In addition to providing an intrinsic evaluation of each of the

models, this also allows us to draw a contrastive comparison between contextual and static encoders, providing insight into both models and demonstrating the method's generalisability to different types of encoders.

With regard to the types of linguistic information that we probe for, in this chapter we explore a semantic probing task, in an effort to investigate what we consider to be the opposite end of the taxonomic—thematic spectrum: a probing task on idiomatic usage. Idioms and multiword-expressions are non-compositional and determining whether the meaning of a phrase is idiomatic or literal is highly dependent on context. We suspect this task to be an example of a semantic problem that is in a sense orthogonal to hypernym-hyponym prediction. As little work has been done on probing idioms, and off-the-shelf idiomaticity probing datasets are not readily available, we leverage an existing idiomatic usage dataset and repurpose it for an idiomatic usage probing task.

It is important to note that in the previous chapter we demonstrated that our method works at the word level. However, many linguistic phenomena, including ones such as idiomatic usage or syntax, are only discernible at the sentence level, with a more complete representation of the context. Hence many existing probing tasks are designed at the sentence level in order to probe for sentence-level information. We highlight that the idiomatic usage task which we explore in this chapter requires our probing experiments to be performed at the sentence level[1].

6.1 Idiomatic Usage Prediction

We first discuss some notable work on modelling idiomaticity to relate the experiments in this chapter to literature on the broader topic of idiomatic usage prediction.

[1] We are conscious that, given that we will be averaging word embeddings to obtain sentence representations, the impact of the information encoded in the norm might be diluted. However, as long as there is a detectable signal, we can claim that the finding is significant.

Multi-Word Expressions (MWEs) are idiomatic phrases, or idioms[2], which are commonly used in all natural languages and text genres (Sag et al., 2002) and are characterised by features such as discontinuity, non-compositionality, heterogeneity and syntactic variability. The dominant view is that idiomatic phrases fall onto a continuum of idiomaticity (Sag et al., 2002; Fazly et al., 2009; King and Cook, 2017), as their meanings are indirectly related to the meanings of their individual constituents (note, for example, the different degrees of semantic opacity in the phrases *kick the bucket* vs. *elephant in the room* vs. *hit the road* vs. *salt and pepper*). Additionally, according to Baldwin and Kim (2010), five sub-types of idiomaticity are recognised: lexical, syntactic, semantic, pragmatic and statistical.

As such, idiomatic phrases are a complex phenomenon, which has been studied with great interest and has been shown to be essential to improving performance of NLP applications such as sentiment analysis (Williams et al., 2015; Spasić et al., 2017), machine translation (Villavicencio et al., 2005; Salton et al., 2014), parsing and word-sense disambiguation (Constant et al., 2017). However, idiomatic phrases still present issues in NLP systems and successfully modelling them has remained an open problem for over a decade.

One reason that the task is so challenging is that new idiomatic expressions can emerge at any time as they are an open set, ruling out any notion of creating an exhaustive list of all expressions for a given language (Fazly et al., 2009). Furthermore, not all occurrences of idiomatic word combinations need to present idiomatic meaning—in certain contexts an idiom can be used in its literal, rather than figurative sense. Studies have shown that literal usage of idiomatic expressions is not uncommon, and disambiguating the usage of an idiomatic expression is not a straightforward task (Fazly et al., 2009; Peng et al., 2014; Salton et al., 2016).

[2]The term MWE frequently encompasses a wide variety of linguistic phenomena such as idioms, compound nouns, verb particle constructions, institutionalized phrases, etc. While the precise definition sometimes differs depending on the community of interest (Constant et al., 2017), in this chapter we use the terms *MWEs*, *idioms* and *idiomatic phrases* somewhat liberally, to mean any construction with idiomatic or idiosyncratic properties. We do not go into too much detail regarding the fine-grained distinctions, as our experiments presented in Section 6.4 are constrained to only one subtype of MWE.

The task of predicting idiomatic usage is typically referred to as *idiom token identification* (Fazly et al., 2009) and it is closely related to the task of word sense disambiguation, as it tackles this problem by aiming to distinguish between figurative and literal instances of potentially idiomatic phrases, given a specific context. Historically, a range of approaches have been developed to model the phenomenon, and the literature reveals a split between research on features that are intrinsic to idioms and more general approaches. Most previous work on idiom token identification deals with building separate models for each given expression, rather than a single general model that could handle all expressions. This is mainly due to the fact that for a long time general solutions were not empirically feasible, given the tandem of limited processing power and the complexity of idioms as a linguistic category.

The earliest per-expression literature explored non-distributional approaches, and initial models were built to leverage features intrinsic to the idiomatic expressions. While work on Japanese idioms showed that features normally used in word sense disambiguation worked well and idiom-specific features were not as helpful (Hashimoto and Kawahara, 2008, 2009), concurrent work on English idioms (Fazly et al., 2009) argued that idioms have distinct canonical forms that distinguish the idiomatic instances of a phrase from its literal instances. These canonical forms were defined in terms of local syntactic and lexical patterns, and could be leveraged for idiom token identification.

Rather than employing idiom-specific features, a significant body of research leveraged discourse and topic-based features. Approaches based on how strongly an expression is linked to the overall cohesive structure of the discourse (Sporleder and Li, 2009) showed that figurative language exhibits less cohesion with the surrounding context than literal language (Li and Sporleder, 2010a,b). Underpinned by this theory, related approaches to the task have explored modelling the behaviour of individual phrases with a focus on discourse and topic

models (Feldman and Peng, 2013; Peng et al., 2014), by framing idiomatic expressions as semantic outliers, thus leveraging an idiom's incongruity with its context.

Some of the per-expression literature also describes work using distributed representations. Peng and Feldman (2017) use word embeddings to analyse the context that a particular expression is inserted in, and predict if its usage is literal or idiomatic, reporting significant improvements over their previous work. Meanwhile, Salton et al. (2016) use Skip-Thought Vectors to create distributed sentence representations and show that classifiers trained on these representations have competitive performance compared with the state of the art per-expression idiom token classification.

However, while effective, modelling the behaviour of individual expressions has its drawbacks: expression-specific models have narrow applicability and aggregating individual models makes systems cumbersome, while providing limited capacity to deal with the problem of disambiguation, and not at all addressing the problem of detecting unknown idiomatic expressions. The preferred approach would certainly be to build a general model, i.e. a single idiom token identification model that can work across multiple idioms, as well as generalise to unseen idioms.

Limited work has been done on such a model: Li and Sporleder (2010a), alongside building their per-expression models, also investigated general models, and found that global lexical context and discourse cohesion were the most predictive features. More recent work (Salton et al., 2017) demonstrated the viability of building a generic idiomaticity model using features based on lexical fixedness. In addition, Salton et al. (2016) also showcased early attempts at addressing some of the issues of per-expression models by demonstrating the feasibility of an approach based on sentence embeddings. Similar to their per-expression models, they use distributed sentence representations generated by Skip-Thought to train a general classifier that can take any sentence containing a candidate expression and predict whether its usage is literal or idiomatic. Their work demonstrated that sentence embeddings

can greatly reduce the amount of discourse history and context required to identify idiomatic usage. By using distributed representations it becomes feasible to build a general classifier with the ability to discriminate idiomatic from literal usage, and the classifier was reported to be as effective as the state of the art data-driven approach at the time.

6.1.1 Probing for Idiomatic Usage

Given that the probing framework forms the methodological basis of this thesis, research most relevant to ours includes work on probing for idiomaticity directly. However, as probing is a relatively recent framework and idioms are a difficult phenomenon to model, little work has been done in this space.

In Section 2.3 we have observed that, while the focus of Salton et al. (2016) was to build an idiom token identification classifier, their pipeline is identical to a typical probing pipeline: sentence embeddings are used as input to a binary classifier that predicts whether the sentence contains a literal or figurative use of a multi-word expression. Salton et al. do not overtly apply the probing framework to their work, yet alongside building a successful idiom identification model, their work undoubtedly shows that an idiom probing task can be successful, indicating that sentence embeddings contain information on the idiomaticity of a sentence—providing the type of inference that is usually drawn from probing work.

More recent work (Nedumpozhimana and Kelleher, 2021) builds upon this notion and reports a set of contextual word-level probing experiments on BERT. The experiments combine a probing methodology with input masking to analyse where in a sentence idiomatic information is taken from, and what form it takes, with results indicating that BERT's idiomatic key is primarily found within an idiomatic expression, but also draws on information from the surrounding context. In addition, there are indications that BERT can distinguish between the disruption in a sentence caused by missing words and the incongruity caused by idiomatic usage.

Meanwhile, Garcia et al. (2021) propose probing measures to assess if some of the expected linguistic properties of idiomatic noun compounds and their dependence on context and sensitivity to lexical choice can be extracted from contextual word representation models like ELMo, BERT and their derivatives. Their probing results on idiomatic noun compounds indicate that idiomaticity is not yet accurately represented by contextual models: while they might be able to detect idiomatic usage, they may not detect that idiomatic noun compounds have a lower degree of substitutability of their individual components when compared to more compositional phrases.

Finally, in our own work (Nedumpozhimana et al., 2022) which is tangentially related to the work presented in this chapter, we have performed sentence-level probing for idiomaticity in BERT. One of our initial observations showed that BERT outperforms Skip-Thought embeddings as used by Salton et al. (2016). In an effort to identify the types of signal that BERT captures in modelling idiomaticity, we used the game theory concept of Shapley Values (Shapley, 1953) to rank the usefulness of individual idiomatic expressions for model training. We found that this metric provides a very good estimate of a given expression's usefulness on the idiom identification task, revealing which idioms are most useful for inclusion in the training set. To better understand the features that make a given expression more or less useful, we have explored idiom-intrinsic properties like fixedness (Fazly et al., 2009), as well as topic-based properties, and have found that providing training data that maximises coverage across topics is the most useful form of topic information. However, our results indicate that there is no one dominant property that makes an expression useful, but rather both fixedness and topic features in combination contribute to an expression's usefulness.

6.1.2 Idiom Benchmarks

In terms of probing for idiomatic usage, popular probing benchmarks such as the ones developed by Conneau et al. (2018) do not include idiomaticity datasets, nor indeed any

kind of explicitly semantic task, as the domain of semantics generally seems somewhat underrepresented in probing work. To our knowledge, only Garcia et al. (2021) have developed a curated idiomaticity probing dataset: they constructed the Noun Compound Senses Dataset for assessing the ability of vector space models to retain the idiomatic meaning of noun compounds in the presence of lexical substitutions and different contexts. The dataset contains a total of 9,220 sentences in English and Portuguese, including variants with synonyms of the noun compound and of each of its components. Other idiom probing work (Salton et al., 2016; Nedumpozhimana and Kelleher, 2021; Nedumpozhimana et al., 2022) relies on existing MWE and idiom datasets, specifically the VNC-tokens dataset (Cook et al., 2008). We present this dataset in detail in Section 6.2.

In addition, several working groups have been established, dedicated to identifying and interpreting MWEs. One of them is PARSEME, with the aim of improving cross-lingual processing of MWEs. While our focus in this thesis is on the English language, it is worth noting that the PARSEME shared task on automatic identification of verbal MWEs (Savary et al., 2017; Ramisch et al., 2018) has had three iterations, offers clear guidelines on annotating verbal MWEs[3], and the group has developed annotated verbal MWE datasets for 27 languages. Among them is also the PARSEME English VMWE dataset, which contains fine-grained word-level verbal MWE annotations on 7,437 sentences (Walsh et al., 2018).

A dataset that has originally been developed outside of PARSEME is STREUSLE (Schneider and Smith, 2015), which stands for Supersense-Tagged Repository of English with a Unified Semantics for Lexical Expressions. The corpus incorporates comprehensive annotations of MWEs and semantic supersenses for lexical expressions. It contains 3,812 sentences and the verbal MWEs in the dataset have recently been additionally annotated for their subtypes, in accordance with the PARSEME guidelines.

[3] https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.1/?page=010_Definitions_and_scope/020_Verbal_multiword_expressions

Another dataset developed outside of the PARSEME working group was developed by Kato et al. (2018), who conducted full-scale verbal MWE annotations on the Wall Street Journal portion of the English Ontonotes corpus (Pradhan et al., 2007), which resulted in a resource containing 7,833 sentences annotated for verbal MWE occurrences, with 1,608 MWE types.

Most recently, the 2022 edition of SemEval introduced a shared task on Multilingual Idiomaticity Detection and Sentence Embedding[4]. The task is aimed at detecting and representing multiword expressions (MWEs) which are potentially idiomatic phrases across English, Portuguese and Galician. They did not constrain the set of phrases to any particular part of speech or syntactic construction, and as such this constitutes a general idiomaticity benchmark. As part of the shared task, training and evaluation datasets have been constructed for each language, containing example sentences with idiomatic and literal usage of the given phrases. The English training set contains 3,328 annotated sentences, however as the shared task has not been completed at the time of writing we cannot refer to any emerging results or findings.

In choosing a dataset for our idiomatic usage probing task, we considered available datasets and tried to establish which one would best lend itself to a probing task. We also hoped to build on related work, aiming to be able to relate our findings to previous work that used these datasets. As the Noun Compound Senses Dataset was not yet available when we started our experiments, we instead opted for one of the verbal MWE datasets. As Conneau et al. (2018) emphasise that a probe should answer a simple, unambiguous question, we required sentence-level annotations with the least amount of ambiguity. We ultimately settled on the VNC-tokens dataset (Cook et al., 2008), which allowed us to keep our framing simple: "does this sentence contain idiomatic usage of a VNC"? To our knowledge this is the only available dataset that exclusively contains VNCs, so in order to preserve the specificity of our experimental analysis, we constrained our experiments to this particular subset of

[4]Task 2: `https://sites.google.com/view/semeval2022task2-idiomaticity`

expressions, which also allowed us to compare to previous work that used the same dataset in their experiments.

6.2 Idiomatic Usage Dataset

We repurpose an existing dataset to serve as data for a new idiomatic usage probing task. Our *Idiomatic Usage* (IU) task is based on the VNC-Tokens dataset (Cook et al., 2008), which is a collection of English sentences containing multi-word expressions called Verb-Noun Combinations (VNC), which can be used either idiomatically or literally. In the cases where these expressions are used idiomatically, they are called Verb-Noun Idiomatic Combinations (VNIC). This includes expressions such as *hit road*, *blow whistle*, *make scene* and *make mark*. Here are some example sentences from the dataset:

- *Bourne **made a mark** on the map .*
- *It is very difficult to **make a mark** in experimental physics these days unless you are already at the top !*
- *As soon as he was out of the bathroom he put on his tracksuit and **hit the road** .*
- *The bullets were **hitting the road** and I could see them coming towards me a lot faster than I was able to reverse .*

The VNC-tokens dataset contains a total of 2,984 sentences with 56 different expressions, with each sentence containing one expression. Each sentence in the dataset is labelled as *Idiomatic usage*, *Literal usage*, or *Unknown*. However, the related literature only makes use of a subset of the full dataset. For consistency and comparability with related work (Peng et al., 2014; Salton et al., 2016; Nedumpozhimana and Kelleher, 2021) we apply the same filtering heuristics: we remove all sentences labelled as *Unknown* from the dataset before running experiments. Furthermore, out of the 56 different idiomatic expressions, only 28 are

Expression	#samples	#idiomatic	ratio
see star	61	5	0.08
hit wall	63	7	0.11
pull leg	51	11	0.22
hold fire	23	7	0.30
make pile	25	8	0.32
blow whistle	78	27	0.35
make hit	14	5	0.36
get wind	28	13	0.46
lose head	40	21	0.53
make hay	17	9	0.53
make scene	50	30	0.60
hit roof	18	11	0.61
blow trumpet	29	19	0.66
make face	41	27	0.66
pull plug	64	44	0.69
take heart	81	61	0.75
hit road	32	25	0.78
kick heel	39	31	0.79
pull punch	22	18	0.82
pull weight	33	27	0.82
blow top	28	23	0.82
cut figure	43	36	0.84
make mark	85	72	0.85
get sack	50	43	0.86
have word	91	80	0.88
get nod	26	23	0.88
lose thread	20	18	0.90
find foot	53	48	0.91
TOTAL:	1205	749	0.62

Table 6.1 VNCs ordered by % of idiomatic usage: number of samples (#samples), number of idiomatic uses (#idiomatic) % of idiomatic usage (ratio).

considered to have a balanced ratio of idiomatic and literal usage in the example sentences, while the remaining 28 idiomatic expression have a skewed ratio. As such, the latter samples are not considered suitable for experiments in the literature. We thus use the subset of 28 VNCs considered to have a balanced ratio, where roughly 60% of instances across the dataset are labelled as idiomatic. After these data preparation steps, the final dataset that we use in our experiments contains a total of 1,205 sentences, of which 749 are labelled as *Idiomatic usage* and 456 are labeled as *Literal usage*, allowing for straightforward binary classification.

A breakdown of each expression in the used dataset is displayed in Table 6.1. The expressions are ordered by increasing order of percentage of idiomatic usage: *see star* is the expression with the lowest percentage of idiomatic usage (8.20%) and *find foot* is the expression with the highest percentage of idiomatic usage (90.57%). The overall percentage of idiomatic instances (regardless of the expression) is 62%.

6.2.1 Choosing the right train and test split

The idiomatic usage task is new to the context of probing, so here we describe the details of how we prepared this dataset for our experiments. In establishing a train and test split we initially considered following the approach of Salton et al. (2016), who aimed to maintain the same ratio of idiomatic and literal usage in both the train and test set for each expression. They split the full dataset into a training set containing roughly 75% of the data and a test set containing roughly 25% of the data, while maintaining the ratio of idiomatic labels and ensuring that instances of each of the 28 VNCs are represented in both the train and test split.

This is a fairly standard approach to evaluating ML systems. However, though the model is not tested on the exact same sentences it is trained on, such a setup still allows it to make predictions on sentences containing phrases it has already seen—which opens up the risk of encountering lexical memorisation (Levy et al., 2015). The presupposition here is that the surface form of a given idiom might carry a signal or informational value for the

classifier. Additionally, as previous work has shown, individual idiom models can be quite successful—once an individual idiomatic phrase's idiomatic behavior has been modeled, it should be fairly easy to disambiguate its usage in new sentences. That being the case, it is quite possible that testing a model on the same VNCs it has seen in training might prime the model to rely on its memory of examples it has already encountered. A powerful classifier would certainly be able to learn individual models of the phrases it has seen in the training data and use that knowledge to classify those same phrases in the test set.

This is not much of an issue if the goal is to evaluate the performance of a VNC classifier. However, our goal here is much more nuanced. We wish to ensure that the probe only learns a general, high-level representation of idiomaticity, that is unrelated to any particular idiomatic phrase, which means we need to remove any confounding factors. With that in mind it becomes clear that the above is not an appropriate way to split our train and test samples. In order for the evaluation results to reflect the probe's model of idiomaticity, rather than its model of any particular VNC, the train and test sets need to be carefully curated. The goal is to probe for the model's idiomaticity information in such a way that, while making a prediction, it would not be able to fall back on its memory or prior knowledge of a given phrase, but would only rely on VNC-independent features to make a prediction. We tackle this issue from two fronts, both the train and test set.

(a) While choosing the test set, we need to consider that different VNCs differ in terms of surface forms, context clues and varying degrees of syntactic flexibility (Fazly et al., 2009). In order to test a general notion of idiomaticity, the probe would need to be tested on a subset of VNCs that it has not seen in training. Having it predict the usage status of only unfamiliar idiomatic phrases would likely force the model to fall back on its general knowledge of what makes an idiomatic phrase, rather than rely on a memory of any specific VNC's property.

(b) In choosing the train set, we also need to ensure that the model attends to general properties of idiomaticity, rather than phrase- or token-specific ones. The surface form of

verb	noun
make	face, pile, hay, scene, mark, hit
pull	leg, weight, plug, punch
blow	whistle, top, trumpet
hit	wall, roof, road
get	wind, sack, nod
lose	head, thread

Table 6.2 Groups of VNCs based on verb constituent overlap.

a given idiom likely has significant informational value for either the encoder or the probe and it is possible that specific constituents of the VNCs might be interpreted as some sort of signal. We have thus inspected the candidate phrases and found that many of the 28 VNCs in the dataset share the same verb constituent, as shown in Table 6.2. In fact, the dataset contains only 7 VNCs that contain "unique" verb constituents: *hold fire*, *have word*, *take heart*, *kick heel*, *see star*, *cut figure*, *find foot*.

This verbal overlap might be interpreted as a signal—were we to include different VNCs containing the same verb in both the train and test set, the probe might recognise the verb and yet again rely on its similarity with what it has encountered during training to make a prediction.

We attempt to mitigate the verbal overlap by populating the train set exclusively with phrases with overlapping verbs, while placing the phrases with unique verbs in the test set. This way the importance of verbs is reduced: an individual verb should not carry as much weight during training because it appears multiple times with different nouns. As such, it does not constitute a strong signal and should not nudge the classifier in either direction. Consequently, more of the representation will be devoted to modelling an abstract idiomaticity, rather than a specific verbal cue.

Coincidentally, satisfying condition (b) also satisfies condition (a), so no additional filtering is needed: the VNCs from the test set do not appear in the training set, and the usage of verbs in the training set is diverse with multiple different VNCs in the train set having the

same verb constituent. We are confident that this is an adequate setup to facilitate the probe extracting a general representation of idiomaticity on both ends (train and test), and so we opt for this split.

As such, our test set includes 7 VNCs, while the remaining 21 are used in training. While this split is not focused on the ratio of training instances, but rather subsets of training instances containing the same VNC, this does mirror the 25%/75% data split employed by Salton et al. (2016). Table 6.3 displays the final train and test split we use in our experiments, as well as a breakdown of specific phrases and their labels in both sets, sorted according to the verbal constituent. Though the 68% ratio of idiomatic phrases in the test set is slightly higher than maintained in previous work (≈68%), we expect the specific choices of VNCs will have a positive effect overall in priming the classifier to use its knowledge of idiomaticity to make predictions.

Additionally, to confirm whether the chosen train and test split is viable and representative of VNC idiomaticity, in parallel with experiments using the train and test split described above, we also perform a second experiment using a form of bootstrapping where we resample the train and test split multiple times by randomly choosing 7 VNCs to be used in the test set, and using the remaining 21 phrases for training. This violates the above-established principle (b) as verbal constituents might be mixed between train and test sets, but still conforms to principle (a), as the model will always be tested on a set of 7 phrases that were not seen during training. Additionally, as we are not fixing the number of samples in the train and test sets, but rather the number of idiomatic phrases (with a varying number of sentences containing each phrase), there will also be slight differences in the ratio of the train and test sample sizes between different runs. However, we find that when the multitude of runs are averaged the true effect comes to the fore—the bootstrapped results mirror the results of the fixed setting, confirming the chosen split[5]. For transparency and completeness, in Section

[5]In fact, a Pearson correlation analysis between the train and test sample sizes and the obtained evaluation scores yields a coefficient no higher or lower than +/- 0.026, showing no correlation.

	Train set			Test set	
VNC	Total	Idiomatic	VNC	Total	Idiomatic
blow top	28	23			
blow trumpet	29	19			
blow whistle	78	27			
get sack	50	43			
get nod	26	23			
get wind	28	13			
hit road	32	25			
hit roof	18	11	cut figure	43	36
hit wall	63	7	find foot	53	48
lose head	40	21	have word	91	80
lose thread	20	18	hold fire	23	7
make face	41	27	kick heel	39	31
make hay	17	9	see star	61	5
make hit	14	5	take heart	81	61
make mark	85	72			
make pile	25	8			
make scene	50	30			
pull leg	51	11			
pull plug	64	44			
pull punch	22	18			
pull weight	33	27			
Total:	814	481		391	268
Ratio:		0.5909			0.6854

Table 6.3 A breakdown of VNCs and idiomatic instances in the train and test split.

6.4 we report results for both setups: Idiomatic Usage Fixed data split (IU_F) and Idiomatic Usage Resampled data split (IU_R).

6.3 Experimental Design

Having established the idiomatic usage dataset and a motivation for the train and test split, we apply the *probing with noise* method analogously to the experiments in Section 5.3, with some modifications. Specifically, we compare the evaluations of thematic GloVe and BERT sentence embeddings.

Applying our method to thematic GloVe and BERT allows us to draw a contrastive comparison between a contextual and static encoder. This provides insight into each model individually, can highlight differences in behaviour, and demonstrates the method's generalisability to additional encoders.

6.3.1 Embedding Models

As highlighted at the beginning of this chapter, the probing tasks we use are framed as classification tasks at the sentence level (see Sections 6.2 and 7.1), so for our experiments we require sentence representations. We use pretrained versions of BERT and GLOVE to generate embeddings for each sentence by averaging the word vectors in the sentence. Despite its apparent obliviousness to word order, this is a common approach to generating sentence representations, is easy to compute and has proven useful in different tasks (Hill et al., 2016).

GloVe As in Chapter 5, for the thematic GloVe embeddings we use the original Stanford pretrained GloVe embeddings[6], opting for the larger common crawl model, which was trained on 840 billion tokens and contains 300-dimensional embeddings for a total of 2.2 million words.

To generate an embedding for the whole sentence we average the word embeddings in the sentence, which yields a 300-dimensional sentence embedding for each sentence. In the rare instance of encountering an out-of-vocabulary word, we generate a random word embedding in its stead[7].

[6] `https://nlp.stanford.edu/projects/glove/`

[7] We have identified 481 unique tokens in the VNC-tokens dataset that do not have a representation in GloVe, 300 of which are relatively infrequent named entities such as *Animorphs, Havilland, MathWorks, Trivers, Xiaolong*, which arguably should not have much impact on the task of idiomatic usage.

BERT For our contextual encoder, we used we use an off-the shelf pretrained version of BERT, specifically the bert-base-uncased model from the *pytorch_pretrained_bert* library[8] (Paszke et al., 2019).

This model generates 12 layers of embedding vectors for each sentence with each layer containing a separate embedding for each individual word in a sentence. To generate an embedding for the whole sentence, our model takes the last layer of the embeddings and averages the word embeddings in that layer. This results in a 768-dimensional embedding for each sentence, which is then used as input to a Multi-Layered Perceptron (MLP) classifier, which labels the input embedding as idiomatic or literal. Note here that we have not specifically fine tuned the BERT embeddings to the idiom token identification problem, but use them as is.

6.3.2 Probing Classifier and Evaluation Metric

As highlighted at the beginning of this chapter, we average the word embeddings in each given sentence. These sentence embeddings are used as input to a Multi-Layered Perceptron (MLP) classifier, which predicts their class labels, and its performance is evaluated using the AUC-ROC score[9]. This evaluation metric is particularly appropriate for this set of experiments as the labels in the VNC-tokens dataset are imbalanced in favour of the positive class (see Section 5.3).

However, referring back to previous literature (Salton et al., 2016), given that we are reporting average scores on the dataset with an awareness of the different VNC's, there is an additional consideration that needs to be made regarding whether to calculate the probe's *macro average* or *micro average* evaluation score.

[8] `https://pypi.org/project/pytorch-pretrained-bert/`

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html`

To calculate a macro average AUC-ROC score we would first calculate it for each VNC and then average these AUC-ROC scores. We had considered using macro average scores because (a) Salton et al. (2016) also reported macro average scores, and (b) macro averaging has the advantage that each expression type will have an equal impact on the overall score, irrespective of the number of instances in the test set that contain that expression. However, its disadvantage is that it does not weight each example in the test set equally. For example, if the expression *make scene* appears in 10 sentences and the expression *hit roof* only appears in 5 sentences in the test set, the performance of a model on a test sentence with *hit roof* will have a bigger impact on the overall AUC-ROC score than the performance of the model on a sentence containing *make scene*.

To account for this, we opt to use a micro average AUC-ROC score. In micro averaging, instead of separately calculating per-expression scores and averaging them, we calculate a score for the full set of sentences in the test set (irrespective of expression). As a result, all test instances have equal weighting towards the final score regardless of the expression in the sentence. We find the micro average score more reliable and relevant to our work, as each test sentence equally contributed in its calculation, further reducing the impact of evaluating performance over individual expressions. We thus only report the micro average AUC-ROC scores for each of the models.

Finally, just like in Section 5.3, to address the degrees of randomness in our method we bootstrap over the random seeds and report the average score of all runs. In the case of our idiomatic usage task, given that the dataset is two orders of magnitude smaller than the dataset in Chapter 5 (as well as datasets to be introduced in Chapter 7), we increase the number of training runs by two orders of magnitude. Specifically, we train the various models 2,000 times where the VNC's in the hold-out test set are fixed (IU_F) and 4,000 times where they are resampled each time (IU_R), and calculate a confidence interval to make sure that the reported averages were not obtained by chance.

6.3.3 Chosen Noise Models

As described in Section 3.3.2, we remove information from the norm by sampling random norm values and scaling the vector dimensions to the new norm. Recall that we only report results pertaining to scaling to the L2 norm. Specifically, we sample the norms uniformly from a range between the minimum and maximum L2 norm values of the respective embeddings in the dataset[10].

To ablate information encoded in the dimension container, we randomly sample dimension values and then scale them to match the original norm of the vector (see Section 3.3.1). Specifically, we sample the random dimension values uniformly from a range between the minimum and maximum dimension values of the respective embeddings in the dataset[11]. We expect this to fully remove all interpretable information encoded in the dimension values, making the norm the only information container available to the probe.

Applying both noise functions together on the same vector should remove any information encoded in it. In this case, the probe should have no signal in the actual embeddings to learn from, which would be akin to training it on random vectors.

Finally, we use the vanilla GloVe and BERT sentence embeddings in their respective evaluations as vanilla baselines against which all of the introduced noise models are compared. Here, the probe has access to both information containers—dimension and norm—as well as class distributions from the training set. However, it is also important to establish the vanilla baseline's performance against the random baselines: we need to confirm that the relevant information is indeed encoded somewhere in the embeddings.

[10]Thematic GloVe: [2.2634,4.2526]
Thematic BERT: [7.4844,11.1366]

[11]Thematic GloVe: [-1.7866, 2.8668]
Thematic BERT: [-5.0826, 1.5604]

GloVe				
Model	$\mathbf{IU_F}$		$\mathbf{IU_R}$	
	auc	±CI	auc	±CI
rand. pred.	.4994	.0015	.4998	.0013
rand. vec.	.4997	.0015	.5	.0013
vanilla	.7485	.0003	.7717	.0022
abl. N	.7445	.0006	.7687	.0021
abl. D	.5012	.0018	.4993	.0015
abl. D+N	.4991	.0018	.5005	.0015

Table 6.4 Idiomatic Usage task experimental results on GloVe, both with fixed (F) and resampled (R) test set. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

6.4 Experimental Results

The experimental evaluation results for the GloVe and BERT idiomatic usage probing task are presented in Tables 6.4 and 6.5. The tables include results for both the setting where the VNC's in the hold-out test set are fixed (IU_F) and the setting where they are resampled each time (IU_R), though this is essentially the same probing task. Recall that all cells shaded light grey belong to the same distribution as random baselines on a given task, as there is no statistically significant difference between the different scores; cells shaded dark grey belong to the same distribution as the vanilla baseline on a given task; and all cells that are not shaded contain a significantly different score than both the random and vanilla baselines, indicating that they belong to different distributions.

The results interpretation here is quite straightforward. In both GloVe and BERT the random baselines behave as expected, with comparable performance in all settings. We can also establish that both GloVe and BERT encode some notion of idiomaticity, as the vanilla baseline significantly outperforms the random baselines in both models.

Comparing IU_F and IU_R: In the idiomatic usage set of experiments it is important to validate our chosen train and test split (see Section 6.2.1) by comparing the respective vanilla

BERT				
Model	$\mathbf{IU_F}$		$\mathbf{IU_R}$	
	auc	±CI	auc	±CI
rand. pred.	.4997	.0015	.4998	.0013
rand. vec.	.4997	.0015	.5013	.0013
vanilla	.8411	.0002	.8524	.0016
abl. N	.8413	.0003	.8532	.0016
abl. D	.4991	.0019	.4978	.0015
abl. D+N	.4999	.0018	.5004	.0015

Table 6.5 Idiomatic Usage task experimental results on BERT, both with fixed (F) and resampled (R) test set. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

performances of IU_F and IU_R. Given that our goal is not to achieve the highest score on this benchmark, but rather to nudge the probe to model a representation of idiomaticity that is unrelated to any given phrase, we expect that this should make the task more difficult for the classifier—allowing it to just memorise the phrases would increase scores, but would not tell us much about the model's encoding of idiomaticity itself.

With that in mind, the results show that in both GloVe and BERT vanilla IU_R significantly outperforms vanilla IU_F. Evidently, the prediction on the task is made more difficult on the curated test split compared to the average of the all resampled splits. Idiomaticity is by no means a simple feature to predict, so we consider this lower performance of IU_F to be a good indicator that the the model is forced to rely on VNC-independent features.

Other than that, in their respective intrinsic evaluations, IU_F and IU_R exhibit the same behaviour in BERT, while there is only one difference in GloVe, namely that ablating just the norm causes a statistically significant drop in performance in IU_F, while this is not the case in IU_R. However in both cases the overarching conclusion about the role of the norm remains the same.

Idiomaticity and the norm: One of the goals of this experiment was to investigate whether the norm encodes any information relevant to the IU task. Based on these results, in

both GloVe or BERT there is no conclusive indication that the norm encodes idiomaticity information on this dataset: in all four scenarios ablating only the dimensions already makes the probe's performance comparable to random. With regard to the lower score when ablating the norm in GloVe IU_F, we suspect this is likely a feature of this particular data split, as the signal is not mirrored in the IU_R. Still, as established in Section 3.7, this is insufficient evidence to infer that the norm encodes the relevant information. While this result leaves us with a number of open questions (see Section 6.5), it is good to confirm that our method is also capable of producing a negative result. It demonstrates that the method does not provide a guarantee that a signal will be detected, but even in this case can prove informative in terms of motivating a post hoc investigation and prompting further questions.

Comparing GloVe and BERT: In terms of differences between encoders, the results show that vanilla BERT significantly outperforms vanilla GloVe in both the IU_F and IU_R scenarios. Evidently, BERT is much better at encoding idiomaticity than GloVe. We suspect this is due to two factors: (a) BERT is a contextual encoder and as such is better suited to modelling the local context necessary to accurately represent idiomaticity in the sentence, and (b) it has a much higher dimensionality, meaning it has the potential to devote more representation space to more complex phenomena.

6.5 Limitations and Conclusion

It is worth noting that while constructing and experimenting with the VNC-tokens dataset we have become aware of some of its shortcomings in the context of our work.

Our main concern is that the dataset is two orders of magnitude smaller than the dataset used in Chapter 5, as well as other typical probing datasets (as used in Chapter 7). While we addressed this by increasing the number of training runs and resampling the train and test set, the preferred scenario is to simply have a larger dataset. Unfortunately, in dealing with an intricate phenomenon such as idioms, considerably-sized corpora are few and far

between. Creating a new dataset from scratch was not feasible given time constraints, but a possible solution we had considered was expanding the dataset by collating additional similar resources. We ultimately decided to forego this step for a number of reasons.

Expanding the VNC-tokens dataset would come with a trade-off in terms of specificity: in its unaltered form, it contains only a single type of verbal multi-word expression, while other available datasets include a wider variety of verbal expressions (Schneider and Smith, 2015; Walsh et al., 2018; Kato et al., 2018) or contain no verbal expressions at all, but e.g. noun compounds (Garcia et al., 2021) instead. Using a broader sample of idiomatic expressions would introduce confounding factors, as not all idiomatic expressions have the same properties (Fazly et al., 2009), have highly varying likelihoods of idiomaticity, and some are exclusively used non-compositionally. Thus, constructing a larger dataset that includes these additional types of expressions would inevitably broaden the probe's search space and complicate the abstraction[12].

Additionally, at this stage this is a relatively older benchmark and there are some indications that it has not been as meticulously crafted as the more recent datasets developed by the PARSEME working group. The dataset also does not control for sentence length, which is a possible confounder[13], but further filtering the dataset to unify sentence lengths would likely render it unusable in its current state. We feel that aligning the dataset with the PARSEME annotation guidelines, cleaning up some of the annotations and updating it with additional examples of sentences containing VNCs in order to better balance the idiomaticity labels would certainly improve its quality. Overall, in spite of our best efforts at mitigating confounders and constructing the right train and test split for our task, we still wonder whether the dataset is simply too small and too imbalanced to truly be useful in a probing scenario.

[12]Note also that due to the difficulty of curating and annotating multi-word expressions, existing resources are within the same size range as the VNC-tokens dataset; concatenating them would certainly increase the absolute size of the dataset, but it would still not even approach the size of the datasets used in Chapters 5 and 7.

[13]While the Pearson correlation coefficient between sentence length and idiomaticity class labels is 0.098, which is quite low, it would still be prudent to only include sentences of comparable length in the dataset.

On the other hand, given that the scope of idiomatic expressions studied is so narrow, the findings may not generalise to other types of expressions beyond VNCs, meaning that the question whether idiomaticity can be encoded in the norm remains an open one. A more exhaustive dataset would have to be curated for a more thorough and general analysis of idiomaticity as such, rather than just idiomaticity of VNCs. We thus emphasise the importance of expanding this work to a wider category of idiomatic phrases and folding in the datasets mentioned above—applying our method to the datasets individually as well as an amalgamation of datasets would provide a more comprehensive and systematic analysis of general idiomaticity encoding and could provide interesting insights. We are committed to exploring this in future work, as well as applying the framework to additional semantic probing tasks.

However, before taking that step, we need to make a final consideration: as opposed to our experiments in Chapter 5 which were performed at the word level, the idiomatic usage experiments have been performed at the sentence level. Given that we simply average word embeddings to obtain sentence representations, it is possible that there might be a signal in the relevant word embeddings, but the move to a higher-order linguistic structure has diluted it enough so as to not be detectable by our method. In other words, we cannot rule out the possibility that perhaps our method does not generalise to the sentence level. Additionally, even if it does, it seems that neither GloVe nor BERT use the norm to encode idiomaticity, which leaves us with an unanswered question: "which information do these thematic encoders store in the norm?" We further pursue this in the following chapter and run additional experiments on the same encoders, but sample a much wider range of sentence-level probing tasks. This allows us to test both whether our method is applicable at the sentence level and whether thematic GloVe or BERT encode any linguistic information in their norm.

In addition, in order to better understand the IU task and dataset, we have run a set of post-hoc experiments and analyses on the IU task. However, we present the results of this analysis alongside post-hoc analyses of the datasets presented in the following chapter. This will allow us to comparatively interpret the findings in relation to the other datasets, and will allow for a more streamlined discussion of the results.

Chapter 7

Probing Static vs Contextual Embeddings: Non-Semantic Tasks

Analogously to experiments in the previous chapter, here we apply our *probing with noise* method to ten existing probing task datasets, as developed by Conneau et al. (2018). The tasks test for different types of linguistic information that span a range of domains such as morphology, syntax and contextual incongruity. This cohort of experiments will provide more general insight into the different types of linguistic information beyond semantics that can be encoded by the norm in thematic embeddings, both contextual and static. Additionally, this allows us to validate that certain types of sentence-level linguistic information can be encoded in the norm of sentence embeddings.

Note that this will make for a comparably short chapter, as all of the necessary groundwork has already been laid: (a) due to the broad nature of these linguistic probing tasks, related work in this space has already been covered in detail in Chapter 2. (b) Given that we use already existing datasets that have been developed specifically for the purpose of probing and have thus been extensively evaluated within this framework and widely adopted by the community, there is no need for any data wrangling nor do there seem to be any intricacies or pitfalls arising from these datasets. (c) Finally, with respect to the application of our

method, the experimental setup is almost identical to what was presented in Section 6.3, with only minor differences in terms of the number of experiments and training runs. Thus, to avoid unnecessary repetition, most of the contents of this chapter will be streamlined and the majority of the focus will be dedicated to the exposition and interpretation of the *probing with noise* results. In addition, here we will also present the subsequent post hoc analysis and experiments, which will include all the datasets presented in this chapter, as well as the IU dataset from Chapter 6.

7.1 Datasets

In our final set of probing experiments we use 10 established probing task datasets for the English language developed by Conneau et al. (2018). In order to inform a discussion on the types of linguistic information that we probe for, we consider these datasets to represent examples of different language domains and group them accordingly. This level of abstraction can lend itself to interpreting the experimental results, as there may be similarities between embeddings trained on tasks belonging to the same domain, which could allow for more general inferences to be made (note that Durrani et al. (2020) follow a similar line of reasoning). The datasets we use are presented below.

- **Surface information**
  - *Sentence Length (SL)* A multi-class classification task where the goal is to predict the length, i.e. number of tokens in the sentence as binned into 6 discreet categories. This is the only one of the 10 dataset where sentences significantly vary in length.
  - *Word Content (WC)* A multi-class classification task with 1000 words as targets, with the goal of predicting which of the target words appears in a given sentence. The data was constructed by choosing the first 1000 lower-cased words occurring

in the source corpus vocabulary ordered by frequency rank from position 2k+1 onwards, and having length of at least 4 characters. Each sentence contains a single target word, and the word occurs exactly once in each sentence.

- **Morphology**
  - *Subject Number (SN)* A binary classification task that predicts the grammatical number of the subject of the main clause as being singular or plural. Only common nouns are considered and only target noun forms with corpus frequency between 100 and 5,000 are considered, and noun forms are split across the train and test partitions.
  - *Object Number (ON)* A binary classification task that predicts the grammatical number of the object of the main clause as being singular or plural. Again, only target noun forms with corpus frequency between 100 and 5,000 are considered, and noun forms are split across the train and test partitions.
  - *Tense (TE)* A binary classification task predicting whether the main verb of the sentence is in the present or past tense. Only sentences where the main verb has a corpus frequency of between 100 and 5,000 occurrences are considered. More importantly, a verb form can only occur in the train or test set, never both.
- **Syntax**
  - *Parse Tree Depth (TD)* A multi-class classification task where the goal is to predict the maximum depth of the sentence's syntactic tree, with possible values ranging from 5 to 12. Since parse tree depth naturally correlates with sentence length, Conneau et al. de-correlated the variables through a structured sampling procedure[1].

[1]They obtained a de-correlated sample by "defining a target bivariate gaussian distribution relating sentence length and sentence depth, setting the co-variance to be diagonal, and sampling a subset of sentences to match this distribution" (Conneau et al., 2018).

- *Top Constituents (TC)* A multi-class classification task where the goal is to predict one of 19 most common top-constituent sequences, plus a 20th category for all other structures of the most common syntactic top-constituent sequences.
- *Coordination Inversion (CIN)*[2] A binary classification task predicting whether the order of two coordinated clausal conjoints in the sentence has been inverted or not. All the sentences in the dataset have coordinated clauses, half are inverted, half are not. The sentences are balanced by the length of the two conjoined clauses, that is, both the original and inverted sets contain an equal number of cases in which the first clause is longer, the second one is longer, and they are of equal length. Also, no sentence is presented in both original and inverted order.

- **Contextual incongruity**
  - *Bigram Shift (BS)* A binary classification task where the goal is to predict whether two consecutive tokens in the sentence have been inverted. The data was constructed by choosing two random consecutive tokens in the sentence, excluding beginning of sentence and punctuation marks.
  - *Semantic Odd-Man-Out (SOMO)* A binary classification task where the goal is to predict whether a sentence occurs as-is in the source corpus, or whether a (single) randomly picked noun or verb was replaced with another word with the same part of speech. The original word and the replacement have comparable frequencies for the bigrams they form with the immediately preceding and following tokens. Both target and replacement were filtered to have corpus frequency between 40 and 400 occurrences[3]. For the sentences with replacement, the replacement

[2]We acknowledge that our categorisation here is somewhat fuzzy as this might not be as directly a syntactic task as the other two. Upon considering the alternatives, syntax seemed like the best fit, though we are conscious that the CIN task could be considered an outlier to a degree.

[3]This range is considerably lower than for the other datasets. The authors motivate this decision with the fact that "very frequent words tend to have vague meanings which are compatible with many contexts". This

words only occur in one partition (i.e. train and test). Moreover, no sentence occurs in both the original and changed versions.

We emphasise that these are 10 separate datasets specifically curated for each task and each of them contains 100,000 annotated sentences in the training set and another 10,000 in the hold-out test set. In all cases, the datasets are balanced across the target classes. We use the datasets as published in their totality, with no modifications[4].

7.2 Experimental Design

7.2.1 Models and Evaluation

As in Section 6.3, we apply the *probing with noise* method to thematic GloVe and BERT sentence embeddings, obtained by averaging the word embeddings in the sentence. The averaged sentence embeddings are used as input to a Multi-Layered Perceptron (MLP) classifier, which predicts their class labels, and its performance is evaluated using the AUC-ROC score. In the case of a multi-class classification task (SL, WC, TD and TC), we calculate the macro average score.

Analogously to Section 5.3, we train the various models 50 times and calculate a confidence interval to make sure that the reported averages were not obtained by chance.

7.2.2 Chosen Noise Models

Yet again, we remove information from the norm by generating random norm values and scaling the vector dimensions to the new norm. We sample the random norms uniformly from

relates to a discussion we covered earlier in the thesis in Section 5.1 relating to distributional generality and the relative frequencies and occurrences of hypernyms and hyponyms.

[4]Full datasets and additional details can be found here: `https://github.com/facebookresearch/SentEval/tree/master/data/probing`

a range between the minimum and maximum L2 norm values of the respective embeddings on all 10 datasets[5].

To ablate information encoded in the dimension container, we randomly generate dimension values and then scale them to match the original norm of the vector. The random dimension values are sampled uniformly from a range between the minimum and maximum dimension values of the respective embeddings on all 10 datasets[6].

7.3 Experimental Results

Detailed experimental evaluation results for GloVe and BERT on each of the 10 probing tasks are presented in Tables 7.1 and 7.2 respectively. Recall that all cells shaded light grey belong to the same distribution as random baselines on a given task, as there is no statistically significant difference between the different scores; cells shaded dark grey belong to the same distribution as the vanilla baseline on a given task; and all cells that are not shaded contain a significantly different score than both the random and vanilla baselines, indicating that they belong to different distributions. Our random baselines behave as expected, having comparable performance across all tasks in both GloVe and BERT. We highlight that in these experiments the random vector baseline (*rand.vec.*) is equivalent to the scenario where both dimensions and norm are ablated (*abl. D+N*). Indeed, we have observed this same behaviour in all of the probing experiments reported in the thesis regardless of the encoder architecture. While the two scenarios are arguably the exact same condition, we include both of them in the results presentation, as it demonstrates a consistent application of our methodology, wherein we consider the *rand.vec.* to be a baseline, and the *abl. D+N* a sense-check of our ablation functions.

[5]Thematic GloVe: [2.0041,8.0359]
Thematic BERT: [7.1896,13.2854]

[6]Thematic GloVe: [-2.5446,3.1976]
Thematic BERT: [-5.427,1.9658]

GloVe										
Model	**SL**		**WC**		**SN**		**ON**		**TE**	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5006	.0013	.4995	.0010	.4996	.0020	.4999	.0023	.4981	.0022
rand. vec.	.4999	.0011	.5006	.0009	.4990	.0022	.4998	.0024	.4997	.0024
vanilla	.9475	.0005	.9974	.0001	.8114	.0014	.7805	.0013	.8632	.0014
abl. N	.9384	.0005	.9940	.0001	.8058	.0016	.7743	.0018	.8594	.0013
abl. D	.5481	.0013	.5040	.0011	.5003	.0022	.4994	.0024	.5013	.0025
abl. D+N	.5001	.0011	.4999	.0008	.4987	.0024	.4994	.0020	.4998	.0021
Model	**CIN**		**TD**		**TC**		**BS**		**SOMO**	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5004	.0022	.5005	.0012	.5005	.0009	.4998	.0022	.4999	.0026
rand. vec.	.4993	.0022	.5002	.0014	.5004	.0009	.4989	.0023	.4991	.0023
vanilla	.5493	.0019	.7799	.0012	.9512	.0004	.5017	.0021	.5291	.0021
abl. N	.5437	.0020	.7689	.0010	.9438	.0004	.5034	.0024	.5235	.0020
abl. D	.5003	.0023	.5137	.0012	.5331	.0013	.4990	.0026	.5005	.0021
abl. D+N	.5004	.0021	.5010	.0013	.4996	.0011	.4996	.0024	.5007	.0019
Key	***Surface Information***				***Morphology***					
	SL: Sentence Length				SN: Subject Number					
	WC: Word Content				ON: Object Number					
	Syntax				TE: Tense					
	CIN: Coordination Inversion				***Incongruity***					
	TD: Parse Tree Depth				BS: Bigram Shift					
	TC: Top Constituents				SOMO: Semantic Odd-Man-Out					

Table 7.1 Experimental results on GloVe models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

GloVe results: The vanilla GloVe vectors outperform the random baselines on all tasks except BS. While this is not surprising, as BS is essentially a local-context task and GloVe does not encode context in such a localised manner, it is still valuable to experimentally confirm that this is the case. In all other tasks, even in cases where evaluation results are quite low when compared to the random baselines, the difference between vanilla and random baseline is still statistically significant, indicating that at least some task-relevant information is encoded in the embeddings.

Having established the vanilla results as a baseline for the ablations, we examine which information container encodes the relevant information: dimension or norm. Generally, the results show that the answers are task-dependent. When it comes to SN, ON, TE, CIN and SOMO, there is a substantial drop in the probe's performance after ablating the dimension container and it immediately becomes comparable to random baselines. Furthermore, performance does not significantly change after also ablating the norm, indicating that no pertinent information is stored in the norm container for these tasks, and that all the information the probe uses is stored in the dimension container.

However, the results for the surface form information probes SL and WC, as well as the syntactic TD and TC probes tell a different story. Once the dimension container is ablated from these vectors, although the performance drops markedly compared to vanilla, it does not quite reach the random baseline performance as observed in the above tasks[7]. These results indicate that for these tasks the relevant information is not contained *only* in the dimension container. Furthermore, when the dimension and norm ablation functions are applied together, this induces a further performance drop, and the resulting performance scores are not significantly different from the random baselines. This indicates that the vectors with ablated dimension information still contain residual information relevant to the

[7]This is true even in the case of WC, where the difference is really quite small, yet still statistically significant. Note that the WC task is a particularly unusual classification task, as there are 1000 possible classes to predict, which could explain the statistical significance of such a small difference.

task, which is removed when ablating the norm, pointing to the fact that the norm contains some of the relevant information *regardless of what is encoded in the vector dimensions*.

We can observe here that in all tasks where at least some task-relevant information is encoded by the vectors (i.e. excluding BS) ablating the norm alone causes a statistically significant drop in performance. Given that we have already encountered this behaviour in Section 5.4 on the hypernym-hyponym results and Section 6.4 with the IU_F results, seeing the same result here further reinforces our interpretation that this finding on its own should not be taken as an indicator that the norm encodes task-relevant information. Given how consistently small the drop is regardless of the task (never larger than 0.1), and given that it does not appear as consistently in the BERT results, this leads us to believe this behaviour is somehow specific to GloVe, perhaps due to an interaction with the noising function.

BERT results: The vanilla BERT vectors outperform random baselines across all tasks, including the BS task, for which GloVe encodes no information, indicating BERT does model word order and takes it into account.

When ablating the dimensions on the SL, WC, SN, ON, TE, CIN and TD tasks, the probe's performance drops dramatically and is comparable to random baselines. It does not change after also ablating the norm, indicating that no pertinent information is stored in BERT's norm container for these tasks. Furthermore, the contexual incongruity tasks (BS and SOMO) show that some of the task information is stored in BERT's norm, as the performance drop when ablating dimensions is not comparable to random baselines, and only reaches baseline performance once the norm is also ablated. The same is true for the syntactic TC task, which is also the only BERT finding that overlaps with the GloVe results, though it seems that BERT stores far less TC information in the norm than GloVe does.

Finally, when ablating just the norm container, only the WC, TD and TC tasks exhibit the small drop in performance observed on most tasks in the analogous GloVe setup. In BERT's case, on the remaining tasks there is no statistically significant drop in performance

BERT										
Model	**SL**		**WC**		**SN**		**ON**		**TE**	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5002	.0006	.4996	.0012	.4995	.0021	.4988	.0022	.5007	.0021
rand. vec.	.5003	.0004	.4997	.0009	.5006	.0020	.4996	.0024	.4993	.0021
vanilla	.9733	.0011	.9820	.0003	.9074	.0008	.8674	.0019	.9135	.0008
abl. N	.9730	.0008	.9783	.0003	.9078	.0008	.8658	.0017	.9118	.0012
abl. D	.5047	.0008	.5013	.0011	.4992	.0021	.5004	.0023	.5007	.0019
abl. D+N	.4997	.0008	.500	.0013	.5006	.0024	.4994	.0024	.4983	.0021
Model	**CIN**		**TD**		**TC**		**BS**		**SOMO**	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5007	.0022	.4999	.0012	.5001	.0013	.5011	.0020	.4990	.0018
rand. vec.	.5014	.0019	.4999	.0012	.5001	.0013	.5005	.0024	.5001	.0021
vanilla	.7472	.0016	.7751	.0016	.9562	.0002	.9382	.0006	.6401	.0013
abl. N	.7492	.0018	.7709	.0016	.9547	.0004	.9371	.0010	.6396	.0017
abl. D	.5049	.0021	.5004	.0013	.5093	.0019	.5560	.0025	.5272	.0020
abl. D+N	.5015	.0035	.5000	.0012	.5001	.0010	.4972	.0035	.4997	.0020
Key	***Surface Information***						***Morphology***			
	SL: Sentence Length						SN: Subject Number			
	WC: Word Content						ON: Object Number			
	Syntax						TE: Tense			
	CIN: Coordination Inversion						***Incongruity***			
	TD: Parse Tree Depth						BS: Bigram Shift			
	TC: Top Constituents						SOMO: Semantic Odd-Man-Out			

Table 7.2 Experimental results on BERT models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded.

compared to vanilla, even in the BS and SOMO tasks where the norm does seem to encode information independent from the dimensions. This shows that there is a certain degree of redundancy between the information in the norm and the dimensions, as even when the pertinent information from the norm is ablated, the information in the dimensions can make up for it.

Ultimately, our experimental results allow us to make a number of general inferences about the norm encoding linguistic information at the sentence level: (a) the norm is indeed a separate information container, (b) on most tasks the vast majority of the relevant information is encoded in the dimension values, but can be supplemented with information from the norm, (c) though the information contained in the norm is not always very impactful, it is not negligible, (d) different encoders use the norm to carry different types of information, (e) specifically BERT stores information pertinent to the BS, SOMO and TC tasks in the norm, (f) while GloVe uses it to store SL, WC, TC and TD information.

7.4 Post-Hoc Analyses and Experiments

Finally, we perform an additional set of supplementary experiments and analyses that improve our understanding of the results and help shape our overall findings. Specifically, we investigate the role of the dimension container by performing a dimension deletion experiment (similar to what was done in Section 5.5), as well as a comprehensive norm correlation study. Note that we perform the post hoc experiments on the 10 probing datasets discussed in this chapter, as well as the idiomatic usage (IU) dataset from Chapter 6. We are able to do this as the tasks are structurally comparable—they are all based on sentence embeddings and probe for sentence-level phenomena. Presenting them as part of the same set of post hoc experiments allows for a streamlined, yet comprehensive analysis, and considering the IU post hoc results in the context of more established datasets with fewer limitations helps us ground and "calibrate" our interpretation of the results.

7.4.1 Dimension Deletion

While the findings that the norm can be used as a carrier of certain types of information are really interesting, our experimental results also show that it is still the case that most of an embedding's information is encoded in the dimensions. With this in mind, we take our experimentation a step further: partially inspired by the work of Torroba Hennigen et al. (2020) who found that most linguistic properties are reliably encoded by only a handful of dimensions, and partially by the intriguing findings from our deletion experiments in Section 5.5, we attempt to roughly identify the degree of localisation of information in the vector dimensions. In staying consistent with the ablational nature of our method and our previous post hoc experiments in Section 5.5, we run another batch of experiments on all our probing task datasets, including the IU task, where we simply delete one half of the vector's dimensions and retrain the probe on the truncated vectors, repeating the process for the remaining half.

It is worth noting here that we are conscious that deleting dimensions reduces the dimensionality of the vector space and inherently changes the norm of the vectors. This serves as an good example for why framing this analysis as a post hoc experiment is important to explicitly acknowledge: it allows us to consider any analysis of dimension deletions and any comparisons with the vanilla baseline as a separate issue from information container ablation analyses. While the ablation functions are used to identify which information container the information is encoded in, doing dimension deletion presupposes that the information is encoded in the dimension container and functions as a test that helps pinpoint where in the dimension container the information is encoded.

The dimension deletion results for the general linguistic probing tasks are included in Tables 7.3 and 7.5, while results for idiomatic usage dimension deletion probing tasks are included in Tables 7.4 and 7.6. In these tables the row denoted *del. 1h* reports the results for deleting the 1st half of an embedding vector, and *del. 2h* reports results for deleting the 2nd

GloVe										
Model	**SL**		**WC**		**SN**		**ON**		**TE**	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5006	.0013	.4995	.001	.4996	.002	.4999	.0023	.4981	.0022
rand. vec.	.4999	.0011	.5006	.0009	.499	.0022	.4998	.0024	.4997	.0024
vanilla	.9475	.0005	.9974	.0001	.8114	.0014	.7805	.0013	.8632	.0014
del. 1h	.9134*	.0006	.9936*	.0001	.7985*	.0019	.7606*	.0019	.8466*	.0016
del. 2h	.9244	.0005	.994	.0001	.8054	.002	.7684	.0021	.8579	.0013
Model	**CIN**		**TD**		**TC**		**BS**		**SOMO**	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5004	.0022	.5005	.0012	.5005	.0009	.4998	.0022	.4999	.0026
rand. vec.	.4993	.0022	.5002	.0014	.5004	.0009	.4989	.0023	.4991	.0023
vanilla	.5493	.0019	.7799	.0012	.9512	.0004	.5017	.0021	.5291	.0021
del. 1h	.5352*	.0018	.7722*	.0006	.934*	.0003	.501*	.0014	.5273*	.0021
del. 2h	.5437	.0017	.774	.0007	.936	.0003	**.5056**	.0022	**.5321**	.0019
Key	***Surface Information*** SL: Sentence Length WC: Word Content ***Syntax*** CIN: Coordination Inversion TD: Parse Tree Depth TC: Top Constituents						***Morphology*** SN: Subject Number ON: Object Number TE: Tense ***Incongruity*** BS: Bigram Shift SOMO: Semantic Odd-Man-Out			

Table 7.3 Experimental results on GloVe dimension deletion models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded. In the dimension deletion experiments the significantly lower score is marked with an asterisk, while the scores marked in bold show an improvement in performance compared to vanilla baseline.

half. When comparing the two deletion conditions for an embedding, in cases where there is a statistically significant difference between them the lower of the two scores is marked with an asterisk. Examining the results reveals some interesting insights.

GloVe deletions: Unsurprisingly, deleting half of the vector generally causes a statistically significant drop in performance when compared to vanilla on most tasks (with some exceptions). However, the drop is also much smaller than might be expected, often very close to vanilla performance. This points to redundancies within the dimensions themselves, indicating that not many dimensions are needed to encode specific linguistic features.

It is to be expected that there would be a drop in evaluation scores regardless of which half of the vector is deleted. However, the observed performance loss is not always comparable

GloVe				
Model	$\mathbf{IU_F}$		$\mathbf{IU_R}$	
	auc	±CI	auc	±CI
rand. pred.	.4994	.0015	.4998	.0013
rand. vec.	.4997	.0015	.5	.0013
vanilla	.7485	.0003	.7717	.0022
del. 1h	**.7737**	.0005	.7553	.0023
del. 2h	.7043*	.0005	.7545	.002

Table 7.4 Idiomatic Usage task experimental dimension deletion results on GloVe, both with fixed (F) and randomised (R) test set. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded. In the dimension deletion experiments the significantly lower score is marked with an asterisk, while the scores marked in bold show an improvement in performance compared to vanilla baseline.

between each respective vector half: on all probing tasks except IU, there is a significantly larger drop in performance when deleting the 1st half of the vector, versus the 2nd half. Typically, we would expect the indices of informative dimensions to be arbitrary, yet this result seems to indicate that GloVe localises the information it encodes in favour of placing more informative dimensions at the beginning of the vector.

However, more surprisingly, in some tasks the deletion causes a statistically significant *improvement* when compared to the vanilla baseline (marked in bold). To be fair, this improvement is quite small in both the BS task, where vanilla GloVe does not actually encode any statistically significant information, and the SOMO task, where the vanilla performance is low to begin with. In the IU_F setup the deletion causes a comparatively large performance spike, but this is not mirrored in the IU_R scenario, so it is possible that it is just a strange artefact of the particular IU_F data split, though it does further reinforce our suspicion that the dataset we used for the IU task has a number of limitations, which makes us question its applicability in this setting.

BERT deletions: Similar to GloVe, deleting half the dimensions causes a significant performance drop in most tasks (except IU). Yet again, the drop is small and quite close

BERT										
Model	SL		WC		SN		ON		TE	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5002	.0006	.4996	.0012	.4995	.0021	.4988	.0022	.5007	.0021
rand. vec.	.5003	.0004	.4997	.0009	.5006	.002	.4996	.0024	.4993	.0021
vanilla	.9733	.0011	.982	.0003	.9074	.0008	.8674	.0019	.9135	.0008
del. 1h	.9385*	.0013	.9757*	.0003	.8728*	.0012	.8319	.0009	.9035	.0008
del. 2h	.948	.0009	.9769	.0003	.8763	.001	.8305	.0009	.9017*	.0007
Model	CIN		TD		TC		BS		SOMO	
	auc	±CI	auc	±CI	auc	±CI	auc	±CI	auc	±CI
rand. pred.	.5007	.0022	.4999	.0012	.5001	.0013	.5011	.0020	.499	.0018
rand. vec.	.5014	.0019	.4999	.0012	.5001	.0013	.5005	.0024	.5001	.0021
vanilla	.7472	.0016	.7751	.0016	.9562	.0002	.9382	.0006	.6401	.0013
del. 1h	.7085	.002	.7699	.0011	9495	.0005	.916	.0006	.6189*	.0017
del. 2h	.708	.0017	.7711	.0012	.9504	.0005	.9116*	.00073	.623	.002
Key	***Surface Information***						***Morphology***			
	SL: Sentence Length						SN: Subject Number			
	WC: Word Content						ON: Object Number			
	Syntax						TE: Tense			
	CIN: Coordination Inversion						***Incongruity***			
	TD: Parse Tree Depth						BS: Bigram Shift			
	TC: Top Constituents						SOMO: Semantic Odd-Man-Out			

Table 7.5 Experimental results on BERT dimension deletion models and baselines. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded. In the dimension deletion experiments the significantly lower score is marked with an asterisk, while the scores marked in bold show an improvement in performance compared to vanilla baseline.

BERT				
Model	$\mathbf{IU_F}$		$\mathbf{IU_R}$	
	auc	±CI	auc	±CI
rand. pred.	.4997	.0015	.4998	.0013
rand. vec.	.4997	.0015	.5013	.0013
vanilla	.8411	.0002	.8524	.0016
del. 1h	**.8668**	.0002	**.8576**	.0016
del. 2h	.8137*	.0003	.8368*	.0016

Table 7.6 Idiomatic Usage task dimension deletion experimental results on BERT, both with fixed (F) and randomised (R) test set. Reporting average AUC-ROC scores and confidence intervals (CI) of the average of all training runs. Cells shaded light grey belong to the same distribution as random baselines, dark grey cells share the vanilla baseline distribution, while scores significantly different from both the random and vanilla baselines are unshaded. In the dimension deletion experiments the significantly lower score is marked with an asterisk, while the scores marked in bold show an improvement in performance compared to vanilla baseline.

to vanilla performance, but its behavior is less predictable than in the case of GloVe. On the ON, CIN, TD and TC tasks, there is no significant performance difference between deleting the 1st or 2nd half of the vectors. The SL, WC, SN and SOMO tasks exhibit a larger performance drop when the 1st half is deleted, whereas the TE, BS and IU task suffer a stronger information loss when deleting the 2nd half. This indicates that there is some localisation happening in BERT as well, but it is not as systematic as it is in GloVe, and it seems to only apply to certain types of information. Peculiarly, the significant performance improvement when deleting the 1st half of the embeddings is repeated in BERT as well, this time in both the IU_F and IU_R splits. Whether this is a genuine reflection of how BERT encodes idiomaticity or just an unusual property of this particular dataset certainly warrants further consideration, and we expand on this in Chapter 8.

A general finding that can be drawn from these experiments is that both GloVe and BERT exhibit a certain degree of information localisation, wherein information seems to be distributed in different ways in the dimension container, with a possible preference for certain dimensions to hold certain information. A logical extension of this line of enquiry is to identify specific dimensions as carriers of specific kinds of information, rather than infer an approximate localisation property. Some work in this direction has already been done (e.g. Karpathy et al. (2015); Qian et al. (2016); Bau et al. (2019); Dalvi et al. (2019); Lakretz et al. (2019); Torroba Hennigen et al. (2020); Durrani et al. (2020)). Our deletion results are certainly consistent with the findings of Torroba Hennigen et al. (2020) and Durrani et al. (2020), who have found that morphosyntactic properties are localised in embeddings, with lower level tasks such as morphology localised in fewer neurons, compared to the higher level task of predicting syntax. In our experiments, we see evidence of localisation in the morphological tasks of TE and SN. Additionally, given how small the drop in the probe's performance is when half the vector is deleted, this indicates these linguistic properties are encoded across a small number of dimensions. In other words, there seems to be a high

degree of information redundancy distributed across the dimensions in a vector, not just between the dimensions and the norm.

7.4.2 Norm Correlation Analysis

While our *probing with noise* experiments have demonstrated which types of information can be encoded in the norm, we wish to better understand the relationship between the norms and the probed information. To this end, we run a post hoc analysis on the norm container. We investigate both the L1 and L2 norms of our embeddings using a Pearson correlation analysis: on each probing task dataset we test the correlation between each vector's norms and the sentence's label. We apply the test to both GloVe and BERT vectors and run it on both the vanilla embeddings and embeddings with an ablated norm container.

This choice does however come with some considerations: the Pearson test is designed to only test correlations between continuous variables, but most of the variables in our experiments are categorical. That said, it is still possible to calculate a correlation coefficient for categorical variables if they are binary, by simply converting the categories to 0 and 1. While we cannot do this in cases such as WC and TC, where there are more than two categorical variables to predict, we can still determine whether there is a statistically significant difference between the categories by using a Kruskal-Wallis test. Unfortunately, this does not quantify the difference in the same way as a Pearson test does, as it does not reveal which of the categories are correlated, nor to what degree, but rather only determines whether any differences in results are significant (similar to an ANOVA test). Hence in Table 7.7 we present the Pearson correlation results, but omit the Kruskal-Wallis results for WC and TC, instead discussing them in the text where appropriate.

We examine the correlation coefficients in light of our *probing with noise* experimental results and find that they support most our findings from Sections 6.4 and 7.3, but notably not all of them. First of all, we must emphasise the finding here that applying our norm ablation

Task	Vectors	GloVe		BERT	
		L1	L2	L1	L2
SL	Vanilla	-0.7278	-0.3758	-0.1564	-0.1039
	Abl. norm	-0.1893	-0.0025	-0.0417	-0.0013
SN	Vanilla	0.0360	0.0268	0.0071	0.0146
	Abl. norm	0.0036	-0.0033	-0.0035	-0.0021
ON	Vanilla	0.0013	0.0008	-0.0736	-0.0583
	Abl. norm	0.0009	0.0013	-0.0181	-0.0010
TE	Vanilla	-0.1152	-0.0571	-0.0542	-0.0413
	Abl. norm	-0.0317	-0.0007	-0.0116	0.0010
TD	Vanilla	-0.0817	0.1908	-0.0415	-0.0251
	Abl. norm	-0.0665	0.0016	-0.0163	-0.0045
CIN	Vanilla	-0.0019	-0.0094	-0.0755	-0.0638
	Abl. norm	0.0029	0.0018	-0.0152	-0.0015
BS	Vanilla	0.0040	0.0002	-0.3866	-0.3238
	Abl. norm	0.0022	0.0006	-0.0978	-0.0005
SO MO	Vanilla	-0.0464	-0.0222	-0.2414	-0.2305
	Abl. norm	-0.0105	0.0000	-0.0420	0.0021
IU	Vanilla	-0.2231	-0.1786	-0.1490	-0.1756
	Abl. norm	-0.0074	0.0276	-0.0397	-0.0167

Table 7.7 Pearson correlation coefficients between the class labels and vector norms for vanilla vectors and vectors with ablated norms.

noise function seems to fully remove the information from both the L1 and L2 norm: the correlation between either norm and the class labels drops to ≈ 0.[8] This is in spite of the fact that we have only scaled the vectors to randomly generated L2 norms, yet the information is also removed from the L1 norm. This provides further support to our initial assumption, as well as our experimental findings, that applying our noising function successfully removes information encoded in the norm along with any distinguishing properties it may have had.

As far as the correlations between the norms and target labels, the data shows that in vanilla GloVe neither norm (L1 or L2) correlates with the task labels for SN, ON, TE, CIN, BS or SOMO, while both norms have a strong negative correlation with SL, and a weak negative correlation with IU labels. Additionally, there is a weak positive correlation between TD and the L2 norm, but not the L1 norm, and a weak positive correlation between TE and the L1 norm, but not the L2 norm. The most highly correlated is SL, confirming that the

[8] Except in GloVe-SL-L1 where the coefficient "only" drops from quite strongly correlated to weakly correlated.

vector norm is used to encode sentence length, as also seen in our experiments in Section 7.3. Finally, the Kruskal Wallis test showed a statistically significant relationship between the labels and the norm for both WC and TC.

When it comes to vanilla BERT, there is no correlation between the norms and labels for SN, ON, TE, TD or CIN. However, both norms have a weak negative correlation with SL and IU, and a moderate negative correlation with BS and SOMO. The latter two are most highly correlated with BERT's norms, and we can take this as an indicator that the vector norm might be responsible for encoding contextual incongruity. Given that this also aligns with our experimental findings in Section 7.3, this is gives further credence to our interpretation that the norm is an information container for these tasks. Regarding WC and TC, the Kruskal Wallis test confirmed a statistically significant relationship between the labels and the norm.

These results align with many of the experimental findings produced by our method. Specifically, in GloVe our method has shown that the norm contains some relevant information for the SL, TD, WC and TC tasks, while for BERT we have found this for TC, BS and SOMO; in our correlation analysis we observe non-zero correlations for these tasks, which aligns with our method's findings.

That being said, there are exceptions, and yet again they center around the IU task. In the case of IU (both GloVe and BERT) the norm exhibits a weak (but non-zero) correlation with the idiomaticity labels, yet our probing method does not provide evidence that the norm encodes idiomaticity information. What makes this unusual is that the correlations with the IU labels, while weak, are comparable to correlations with TD or SOMO, which do produce a signal when examined by our method. This could potentially indicate that the relevance of the correlation strength is task-dependent—while for certain tasks fairly weak correlations align with a signal in our method, in others this correlation is too weak to translate into a detectable signal. On the other hand, this could be a sign that other factors are at play—we suspect that in this case this misalignment between our method and the correlation results further hints at

the imbalanced nature of the IU dataset and its limitations, where the correlation between the norm and IU labels is possibly spurious. If this is the case, it shows that our method is more robust than the correlation analysis alone, as it is not so "easily fooled" by this spurious correlation.

However, beyond simply confirming observations we have already made in previous experiments, this post hoc correlation analysis can also help us better understand how the respective linguistic phenomena are encoded in vector space. This is revealed by interpreting the positive or negative sign in front of the correlation values of the vanilla embeddings. We interpret them as follows: for the binary classification tasks, a positive correlation coefficient means that a longer norm indicates the positive class, while a negative correlation means that a longer norm indicates the negative class. For example, on the SOMO task in BERT, the negative correlation coefficient means that sentences containing an out of context word (here considered the positive class) are positioned closer to the origin relative to sentences that do not contain it. In multi-class tasks such as SL or TD, which have an ordinal variable as their target class, a positive correlation coefficient means that a longer norm indicates a larger target level, while a negative correlation means that a shorter norm indicates a smaller target level. For example, the negative correlation coefficient on the SL task in GloVe indicates that the norm of longer sentences is shorter, meaning they are positioned closer to the origin.

Based on this principle, examining the GloVe column in Table 7.7 shows a negative correlation between the norm and the IU, SL and TE labels, meaning that the norm of longer sentences is shorter, that the norm of sentences containing idiomatic usage is shorter, and that the norm of sentences containing a verb in past tense is shorter, positioning them relatively closer to the origin. Conversely, the positive correlation in TD indicates the opposite relationship, meaning that the deeper the syntactic parse tree, the further away the sentence is positioned from the origin. On the other hand, the BERT column contains no notable positive correlations, but only shows negative correlations for the SL, BS, SOMO and IU

task, meaning that, for example, sentences containing words with swapped positions or an odd-man-out are found closer to the origin.

7.5 Conclusion

In this chapter we have applied our *probing with noise* method to 10 existing sentence-level probing task datasets that belong to a number of linguistic domains. Generally, our findings offer both negative and positive results, confirming that our method is applicable at the sentence level and that the norm of thematic embeddings can encode certain types of linguistic information in the norm.

More specifically, we have found that, while both encoders store the majority of sentence-level linguistic information in their dimension containers, they sometimes supplement that encoding by storing information in their norms, but the type of information differs depending on the encoder. Specifically, BERT seems to mostly store information pertinent to contextual incongruity in the norm (with some syntactic information included), while GloVe mainly uses it to store syntactic and surface level information.

We also note that, while the differences in the scores of the various probes do not seem as impactful as what we have observed on the word-level task in Chapter 5, given that here the probing was done at the sentence level, we suspect the signal would likely have been stronger if examined at the word level, where it would not be diluted by averaging.

In addition to the results showing which dimension container encodes which linguistic information, our post hoc analysis has supplemented our interpretation of these results and has given us a better understanding on how the information is encoded in each respective container: the deletion experiments have revealed that relevant information tends to be localised in the dimension container, with GloVe exhibiting a blanket preference for the first half of the vector dimensions, and BERT showing localisation tendencies only for certain language tasks. Furthermore, our norm correlation analysis has shown that BERT sentence

vectors are positioned relatively closer to the origin of the space if they contain contextual incongruity or idiomaticity. GloVe also positions sentence vectors with idiomaticity closer to the origin, but positions sentences with deeper syntactic parse trees further from the origin. It also seems that in both GloVe and BERT longer sentences are positioned closer to the origin.

In the following chapter we build a discussion around these findings, as well as all the other findings presented throughout this thesis. We will take a step back and examine them in their totality, in order tie together any loose threads and build a more coherent narrative within the larger context of the thesis.

Chapter 8

Discussion

Given the large number of results obtained across the experiments and post hoc analyses presented throughout this thesis, specifically Chapters 5, 6 and 7, a large number of variables have been introduced and explored, resulting in a lot of moving parts. In this chapter, we take a moment to tie together relevant findings and ponder their implications. One of the driving hypotheses of our work was that the norm of different encoders can carry different types of linguistic information. Our experiments have provided a number of insights into which encoders encode which types of information in the norm, and in what way the information is encoded. In this chapter we will first discuss these findings in more detail and relate them to existing findings in the literature. We will then take a broader look back at the thesis and discuss differences we have observed between contextual and static encoders, as well as differences between the taxonomic and thematic embeddings. To close out the chapter, we will discuss the limitations of our research and lay out plans for future work.

Sentence Length (SL) Both our *probing with noise* experimental results and the norm correlation analysis in Section 7.4.2 have uncovered a very strong signal that GloVe's norm encodes sentence length. However, given that we obtain sentence representations by averaging the word embeddings of words in the sentence, there is no way an encoder such as

GloVe could be directly encoding a sentence length property when its goal is to produce word-level representations. Yet given how strong the signal is, it cannot be dismissed as an outlier. For an explanation, we look for support in related literature: Adi et al. (2017) have also examined the relationship between sentence length and the norm of word2vec embeddings. They have shown that the embedding norm decreases as sentences grow longer, which is consistent with our findings. They suspect this plays a role in the high evaluation scores of their sentence length probing tasks, and offer a mathematically-informed interpretation of this unexpected correlation:

> Consider the different word vectors to be random variables, with the values in each dimension centered roughly around zero. Both central limit theorem and Hoeffding's inequality tell us that as more samples are added, the expected average of the values will better approximate the true mean, causing the norm of the average vector to decrease. We expect the correlation between the sentence length and its norm to be more pronounced with shorter sentences (above some number of samples we will already be very close to the true mean, and the norm will not decrease further), a behavior which we indeed observe in practice. (Adi et al., 2017, pages 6-7)

This tendency of the decreasing norm of the averaged vector is a logical explanation of why the norm of a sentence representation derived from averaged word2vec word vectors correlates with sentence length. Our experiments and norm analysis on the SL task confirm that these findings hold on GloVe: the negative correlation shows that the longer sentences have shorter norms and vice-versa. However, we note that this is not a property that is inherently encoded by the embedding models, as the word embeddings have no way of extracting and storing the sentence length of a test sample based on the data from their original training corpora. Rather, this is a property of the averaging approach to generating

sentence embeddings, and as such does not reflect information stored in these particular embeddings during their training.

Still, this fact does not undermine the inference that the norm is capable of encoding some kind of information about the representation, nor does it undermine the fact that our experiments demonstrate that a probing classifier can access this information and use it to make predictions, even when the norm value is not explicitly provided, both of which are important findings to take away from our results.

That being said, our results do not show that sentence length being stored in the norm of averaged word embeddings generalises to BERT embeddings: the correlation between SL labels and the norm is much weaker in BERT, to the point where our probe is not able to detect a significant signal—ablating the dimensions achieves results comparable to random, even though the correlation between SL labels and the norm is non-zero. Given that Adi et al.'s explanation is mathematically grounded, it should hold regardless of which type of encoder is used to produce the averaged sentence embeddings. However, the correlation between sentence length labels and BERT's norm is dramatically weaker that the same correlation in GloVe, indicating that this mathematical reasoning does not apply to BERT. The most likely explanation for this would be due to BERT using the GELU activation function (Hendrycks and Gimpel, 2016), which results in vector values that are not centred around zero, meaning that adding more samples (words) to the calculation of the averaged sentence representation won't push the values towards zero, thus weakening the correlation between the sentence length and norm.

Syntactic Information (TD and TC) We observe another strong signal in GloVe embeddings, which indicates that GloVe's norm encodes syntactic information. Interestingly, while the correlation between the TD labels and the L1 norm is negligible, the TD labels' correlation with the L2 norm is stronger by two points. Our probing experiment supports this finding, as ablating the dimensions does not cause a drop to random-like performance,

indicating that the probe is learning the TD information from the norm[1]. While less pronounced, we observe the same matching signal on the TC task, where our dimension ablation experiments show above-random performance, and our Kruskal-Wallis test indicates that the relationship is statistically significant.

This is another finding that, in principle, finds support in the literature: recall that Hewitt and Manning (2019) have investigated in detail how syntactic parse trees are encoded in vector space. Their findings demonstrate that it is possible to recover parse trees from contextual sentence representations, showing that the squared L2 norm corresponds to the depth of the word in a parse tree. However, an important aspect of their work does not align with our findings: they performed their experiments on BERT and ELMo embeddings, and while we can see that GloVe encodes the same syntactic information in the L2 norm, we do not detect the same signal in BERT.

Our correlation study does not indicate a correlation between the TD labels and either of BERT's norms, nor is our probe able to achieve above-random performance when the dimensions are ablated, indicating that TD information is not encoded in BERT's norm. We suspect that the findings of Hewitt and Manning (2019) are dependent on the particular probe they use, whereas we employ an altogether different probe in our experiments. While our probe is able to detect the norm's relevance to encoding tree depth in GloVe embeddings, it might not be capable of recovering the encoding of the parse tree in the norm of contextual embeddings as identified by Hewitt and Manning. However, we suspect the more salient difference is that they predict the depth of an individual word in a sentence given a contextual word embedding as input, while in the TD task our goal is to predict the maximum depth of a sentence's tree using an averaged sentence embedding as input. We suspect that this difference in the pipeline is the main reason we are not able to reproduce their result, as it is

[1]Recall that the TD dataset is decorrelated in terms of sentence length and tree depth, meaning that any sentence length information encoded in the norm of GloVe sentence vectors should not affect the norm's correlation with the TD labels.

likely that, even if the norms of individual word embeddings did encode their depth in the parse tree, this effect is lost when the vectors are averaged to obtain a sentence embedding.

Given this discrepancy, another question naturally arises: would Hewitt and Manning's probe lead to the same result in GloVe embeddings? They themselves do not provide an answer as, while they do compare against certain baselines, none of them include GloVe or similar architectures. Yet we suspect their work would not be reproducible in a setting such as ours—their probe requires the input of a contextual embedding, which generates a different word representation for the same word in different contexts. Given that GloVe cannot provide such representations, it is highly unlikely that their word-level probe would uncover the word's depth in the parse tree.

Contextual Incongruity (BS and SOMO) When it comes to contextual incongruity information, we observe a strong signal in BERT, but not in GloVe. BERT's norm contains information relevant to the BS and SOMO tasks—correlations are considerable and the probing experiments reveal a significant amount of information is left over after dimensions are ablated.

It is notable that specifically BS and SOMO exhibit this signal—we see these as related tasks, given that both violate the local context of the affected words. To expand on this, consider a hypothetical scenario where an LSTM language model encounters two words with swapped positions, or a word that is an odd-man-out: it is likely that at that time-step it would measure high perplexity, though the overall context of the sentence would likely be fine—in other words, these tasks capture local contextual incongruity. We know that BERT is a contextual encoder which is known for its capacity to accurately model short-distance dependencies and word co-occurence probabilities, concepts which strongly relate to local contextual incongruity. We also know that BERT's self-attention uses the vector norm to control the levels of contribution from frequent, less informative words (Kobayashi et al.,

2020). We suspect that this ability is related to the fact that BERT's norm encodes some contextual incongruity as well.

If word co-occurrence frequencies are indeed the signal that BERT is capable of encoding, it is undoubtedly using the norm to supplement its encoding of contextual incongruity. This is evocative of how some embedding models position stop words near the origin. As stop words co-occur with everything, and are not sensitive to context or topic, they need to be more or less equidistant to everything else in the space. Analogously, when it comes to sentences that contain contextually incongruous phrases, it seems BERT is unable to position them close to existing contexts. These phrases violate the local context and so BERT cannot predict which context they belong to, falling back to positioning them closer to the origin, to be closer to all contexts. Hence, BERT sentence embeddings with a bigram shift or an odd-man-out end up distinguished in vector space by their relative distance from the origin.

Idiomatic Usage (IU) Our findings on the IU task in Chapter 6 demonstrate a similar effect as in the SL task: while the correlation coefficients between both GloVe's and BERT's norm and the IU labels are considerable (they are in the same order as, for example, TC in GloVe, or SOMO in BERT), our probe does not seem to be able to leverage this information from the norm, as ablating dimensions immediately yields random-like results. This scenario serves as another example of why the post hoc analyses are complementary to our method: it demonstrates that a correlation analysis on its own would not be a good indicator of whether the norm actually encodes information relevant to the task. In isolation, the correlation coefficient would have led us to believe that there may be some idiomaticity information encoded in the norm. However, this has not been confirmed by our *probing with noise* method, which when used in conjunction with the correlation analysis can offer more nuanced insight.

Admittedly, we are somewhat puzzled by the result that our probe cannot retrieve the IU information seemingly stored in the norm, due to our understanding of the nature of idiomatic phrases. As outlined in related literature (see Section 6.1), many researchers agree that

idiomatic phrases are at least partially defined by how strongly they are linked to the overall cohesive structure of the immediate discourse. Based on this understanding, our intuition is that the IU task should behave similarly to the BS and SOMO tasks which deal with contextual incongruity: using a phrase such as *spill the beans* in a sentence that in no way relates to food, cooking, kitchens or shops would surely have a similarly confounding effect on the word co-occurrence statistics of the sentence as any example of an odd-man-out from the SOMO dataset. Indeed, this reasoning aligns with the findings of Nedumpozhimana and Kelleher (2021), who found indications that BERT can distinguish between the disruption in a sentence caused by missing words and the incongruity caused by idiomatic usage. Based on this, we would be inclined to consider that the IU task might also be a contextual incongruity task, yet our results indicate the opposite. We question whether BERT truly does not encode idiomaticity information in the norm or if there are other factors at play. In search of answers, we consider our dimension deletion results (see Section 7.4.1) for further insight.

The dimension deletion post hoc experiment on the IU task shows that deleting half the vector in both GloVe and BERT causes a significant performance spike. This is baffling in and of itself, especially given the significant differences between the GloVe and BERT architectures. These differences are also most evident when observing their respective performance on the BS and SOMO tasks—GloVe does not perform well at all on contextual incongruity and does not use the norm to encode this information. However, interestingly, these two tasks in GloVe are also the only other scenarios where we observe a statistically significant performance spike when deleting one half of the vectors. This could possibly hint at a relationship between the incongruity and idiomaticitiy tasks, at least in GloVe. However, if a relationship were there and the IU task were truly also a contextual incongruity task, then vanilla GloVe should arguably be much worse at encoding it than it is; we would expect it to be closer in performance to BS and SOMO. Meanwhile, vanilla BERT strongly outperforms vanilla GloVe on the IU task, which lends some credence to the interpretation

that the contextual awareness and the ability to model incongruity, which GloVe lacks but BERT excels at, is what improves the model. However, if we accept that premise, then we are left with the question of why this does not apply to the IU task as well in the sense that the information is not at least partially reflected in the vector norm.

It is also worth considering the findings from our related work (Nedumpozhimana et al., 2022) which indicate that there is no one dominant property that makes an idiomatic expression useful for the probe, but rather that both intrinsic and topic features in combination contribute to an expression's usefulnesss. This speaks to the complexity of idiomatic usage in language, and suggests that BERT achieving state-of-the-art performance on the task of general idiom token identification could be attributed to its ability to combine multiple forms of information (syntactic, topic, contextual incongruity, and so on) rather than simply focus on a specific information type as the explanatory signal for idiomatic usage behaviour across all expressions.

Given that we cannot find an answer without further research, we take a step back and consider our experimental results: compared to all other tasks, most of the results we have observed on the IU dataset behave like unusual outliers that are difficult to explain. This can either be due to strong confounding factors at play that we are not aware of, or, perhaps significantly more likely, this is further evidence of our suspicion that the dataset is just not well-suited for this type of analysis (as already discussed in Section 6.5). And while we have learned that vanilla BERT—a contextual encoder—is better at the task than GloVe—a static encoder—the question whether idiomaticity can be encoded in the norm remains an open one.

Differences Between Contextual and Static Embeddings In Chapters 6 and 7 we applied our method to two thematic encoders and tested them on a thematic semantic task (IU) as well as other, non-semantic linguistic tasks. While our results have provided insight into which types of linguistic information can be encoded by the norm, they have also revealed some

notable differences between static and contextual encoders. Here we shift our perspective and focus on discussing our results in light of the differences found between contextual and static embeddings, as exemplified by BERT and GloVe.

In terms of the types of linguistic information that are captured by the norms of the different encoders, it seems that they are both capable of capturing surface-level information like sentence length and word content. However, that claim is far better supported by our results for GloVe than for BERT, which only shows such indications in the post hoc analyses, rather than our main experiments. They also seem to both encode a degree of syntactic information in the norm, although it seems BERT also stores less of it when compared to GloVe[2]. They both produce the same result on the idiomatic usage task, displaying an above-zero correlation with the class labels, but do not produce a signal when probed with our method. The clearest difference between them is the difference in encoding contextual incongruity information. The GloVe model is not really designed to capture this type of information at the sentence level, and thus not even its vanilla iteration achieves above random baselines. While there is a statistically significant improvement over the baselines on the SOMO task, the improvement is still quite minor. In comparison, vanilla BERT is quite capable of encoding both these tasks, and also shows indications that this information is partially encoded in its norm.

These differences are likely due to a combination of factors: a contributor is the fact that the sentence representations are an average of individual word embeddings, which could be diluting the signal to a degree. Additionally, it is possible that BERT's higher dimensionality provides it with additional capacity to store certain types of information directly in its dimensions, while reserving the norm for higher-level information types such as contextual incongruity. In contrast, GLOVE is a static encoder and exhibits no indication

[2]Recall that both our method and post hoc correlation analysis supports the finding that GloVe encodes TD and TC information in the norm, for BERT we only find such evidence for TC, but not TD.

that it stores this information in the norm, or indeed any real ability to accurately model these phenomena at all, and instead uses the norm to store surface-level and syntactic information.

Additional differences are revealed by our post-hoc deletion experiments, which show that the two types of encoders differently localise the information in their dimension container[3]. GloVe exhibits an overall tendency for storing more pertinent information in the first half of its vector on all tasks (except IU), meaning that the information loss is less severe if the second half of the vector is deleted. BERT does not follow a similar pattern, nor really any pattern at all. It seems that in BERT this localisation property is task-dependent, as it is not exhibited on all tasks (only in SL, WC, SN, TE, BS, SOMO and IU). BERT also does not seem to have a tendency towards storing relevant information within either half of the embedding—in cases where there is a significant difference, in four tasks the 1st half stores more information (SL, WC, SN, SOMO), while in three tasks the 2nd half stores more (TE, BS and IU). There also does not seem to be a relationship between BERT's localisation tendency and whether the task-relevant information is stored in the norm—there are cases (a) where there *is* a significant difference between deletion scores and information *is* stored in the norm; (b) where there *is* a significant difference and information is *not* stored in the norm; and (c) where there is *no* significant difference and information is *not* stored in the norm.

The observed localisation properties of BERT embeddings are within the boundaries of the expected. Recall our consideration of the impact of dimension shuffling in Section 3.3.1: in principle, dimension "semantics" are arbitrarily assigned, so there is no reason why dimension 1 would need to contain a specific type of information, when it could just as well be dimension 257, as long as all the values within the vector remain the same and their indices consistent throughout a dataset. In fact, given this arbitrariness, it is quite possible that the splits we have observed in BERT are indeed there just by chance. Were the BERT model retrained we might observe significant differences on different halves, or even

[3]Recall that we interpret the results as: if there is a statistically significant difference between deleting the 1st half or the 2nd half of the vector, this indicates that information might be localised. If the difference is not statistically significant, then information is likely not localised.

on different probing tasks. And while more research is needed to confirm this property in BERT, our work does reveal somewhat surprising regularities in how GloVe encodes this information. It seems that GloVe consistently places more informative dimensions in the first half of the embedding. It is likely that, given the large number of experiments, the most likely reason for this pattern is that it is caused by experimental variation and is simply accidental. However, given that it is a consistent and statistically significant signal across all our deletion experiments, it does makes a potentially interesting topic for future analysis. A potential hypothesis to explore in future work is that this might be a consequence of the matrix factorisation methods specific to calculating the GloVe embeddings, which are not featured in BERT's architecture.

Finally, while these are significant differences in how BERT and GloVe store information in their dimension container, this is overall insufficient evidence to claim that these findings would generalise to other static and contextual encoders, and more research is needed to confirm such a relationship.

Differences Between Taxonomic and Thematic embeddings In addition to thematic GloVe and BERT embeddings, in Chapter 5 we have also applied our method to taxonomic SGNS and GloVe embeddings, and tested them on the taxonomic probing task of hypernym-hyponym detection. Having thus examined both ends of the taxonomic—thematic semantics spectrum to some degree, we shift our perspective again towards a joint discussion of their differences.

In terms of our semantic probing tasks, our hypernym-hyponym probing experiment has shown that taxonomic embeddings—both SGNS and GloVe—contain a significant amount of hypernym-hyponym information in their norms, while their thematic versions do not. Meanwhile, our IU experiments have shown that thematic embeddings—both GloVe and BERT—exhibit some correlation between the norm and idiomaticity labels, but our method cannot confirm that their norm does encode this semantic feature.

In terms of our non-semantic probing experiments, we have answered a question posed in Section 5.7: what other types of information are encoded in thematic GloVe embeddings? We now know that thematic GloVe mainly encodes syntactic information, as well as some word content information, while BERT on the other hand encodes mainly contextual incongruity information, as well as some syntactic information.

This seems to suggest that the norm of specialised embeddings can be leveraged to encode the specialised property—e.g. taxonomic embeddings encoding taxonomic information in the norm. Meanwhile, given that the thematic embeddings we have used were not in any way specialised for a specific type of information, the norms of thematic embeddings have been shown to contain a variety of non-taxonomic information, spanning from surface level, through syntactic and contextual, with an inconclusive hint of idiomaticity[4].

Notably, the impact of the information present in the norms of thematic embeddings seems to be far weaker than in taxonomic embeddings[5]. Granted, it is likely that this is simply due to the jump from word-level to potentially less precise sentence-level representations. However, we ponder the possibility that it might be due to the lack of specialisation in the thematic embeddings we have used. "Generic" embeddings such as BERT often achieve state of the art results on many tasks, with no in-domain specialisation. As such, it is possible that their norm cannot be dedicated to encoding one type of information really well, but is rather spread more thinly across language domains. Then, instead of a collection of generic non-taxonomic information, idiomatic usage might be more saliently encoded in the norm of embeddings that were somehow specialised to encode this type of semantic information. Analogously, perhaps an encoder specialised to retrieve a sentence's syntactic structure might

[4]With the caveat that the IU results might require a replication experiment based on different datasets to confirm the finding, we are careful about making sweeping statements here, but consider this to be a sound basis for further research.

[5]Recall that, even while results are statistically significant, there are large differences between the performances of embeddings without any dimension information: taxonomic GloVe achieves an AUC-ROC score of $\approx$0.66 on the hypernym-hyponym task, while the best performing thematic model reaches a score of only $\approx$0.56, indicating that more pertinent information is stored in the taxonomic model's norm.

be more inclined to store this information in its norm than a generic BERT or GloVe model would.

If we accept this as a likelihood, then presumably the opposite should also be true—specialised embeddings would not use the norm to store information that falls outside of their domain of "expertise". Following this reasoning, it would be interesting to investigate additional questions: Is any general sentence-level linguistic information encoded in the norm of taxonomic GloVe embeddings? Do they encode the same information as thematic GloVe, some other information, or none at all, instead focusing on encoding hyponym-hyponym relations? How do their vanilla iterations perform on general linguistic tasks, compared to non-specialised embeddings?

While it may seem that we have all the necessary ingredients to perform such a study, we expect that using the WordNet random walk embeddings created in Chapter 4 to study these questions would be futile. Given the way they were trained—using pseudo-corpora obtained via random walks of the WordNet taxonomy—there would be no way for the encoders to extract any kinds of sentence-level linguistic information, as the pseudo-corpora feature no natural language morphology or syntax. Performing an evaluation of embeddings on a task based on predicting linguistic properties of natural sentences would be a misguided attempt at judging how well apples compare to oranges.

Admittedly, we do acknowledge this as merely an educated guess and cede that empirical proof is needed to confirm that this would be the case, though we would be quite surprised to see any additional type of information encoded in the norm of our WordNet random walk taxonomic GloVe or SGNS embeddings. That said, this is a key dimension we are missing here in order to make a full empirical comparison of taxonomic and thematic embeddings. We consider this to be one of a number of limitations of our work, which we discuss in Section 8.1.

Probing with noise While the *probing with noise* method and supplementary analyses have provided a number of insights into embeddings, most notably that the norm of embeddings can encode certain types of linguistic information, a common criticism often aimed at exploratory empirical work, which applies here as well, is one concerning the impact of the findings. Having shown that some amount of information can be encoded by the vector norm, the question that often follows is: How is this knowledge relevant to the wider community and what is the applicability of these findings? While not all, some of the signals discovered in our work could be considered relatively weak, and given that there also seems to be some redundancy between dimension and norm information, this rightly puts into question the relevance and applicability of the results. While they are valid concerns, here we wish to expand on this discussion and clarify certain finer points.

Most importantly, we reiterate that this is an exploratory, empirical study of the geometric properties of different types of embeddings. We have extended the existing probing framework and devised a method that allows us to peek deeper into the black box of language representations, with the goal of expanding our understanding of the way certain models encode information. We believe our results have improved our understanding of the mechanics of vector space models and provided insights relevant to the domain of model interpretability. Just as importantly, it has allowed us to reframe our understanding of work in this space, showing that identifying the information containers relevant for the target information is a necessary prerequisite step to doing any research on embeddings, whether it be a post hoc analysis or further experimentation involving one of the information containers.

Following this framework allows us to determine possible confounders and allows for an awareness of the impact of performing certain operations on vectors, if such operations are a part of the research. For example, without this awareness, any work involving operations where vectors are normalised—such as when employing cosine similarity, as discussed in Section 3.1—could result in unwanted information loss. With our method, it is easier to

identify what information is lost, which allows for an informed decision regarding whether the information loss is relevant to the research at hand. Furthermore, performing research on things like dimension selection or analysing individual neurons to explore information encoding in the dimension container, without allowing for this prerequisite step of identifying where the information is encoded, gives no consideration to the fact that the pertinent information could be located outside of the studied container. Indeed, certain findings in related work claim that (morpho)syntactic information is encoded in a subset of dimensions (Torroba Hennigen et al., 2020; Durrani et al., 2020), but do not give due consideration to the norm in their settings. With our newfound understanding of the norm's relevance in encoding (morpho)syntactic information, it is important to ask whether the information encoded in this subset of dimensions is the same information that is encoded in the norm. If it is, it might be possible that there are dependencies between the information containers and that the performance of the dimensions relies on information encoded in the norm. We caution that this is a prudent consideration to make, and possibly control for, as not distinguishing the contribution of the different information containers runs the risk of simply ignoring the contribution of the norm container, leading to incomplete results interpretation. Even worse, any novel probe that modifies the dimension container might have a negative impact on the encoding of information in the norm container, resulting in information loss.

That being said, some might argue that, while significant, some of the signals we have uncovered are quite weak to be truly relevant and the impact of the information loss would likely be negligible. However, we argue that finding any recoverable information in the norm is actually a strong and relevant result, no matter how weak the signal might be. It is a well known fact that most of the information in an embedding is encoded in the dimensions—indeed, we have shown that for many tasks *all* the information is encoded exclusively in the dimension container and the norm holds no extractable information pertinent to the task. This is to be expected as this is how vector representations of linguistic units were

envisaged in the early days, as each dimension representing some kind of information, with no consideration for the norm (consider the logic behind one-hot encodings). Next to anything between 300 and 700 dimension values, there is no reason why the norm should have to be a relevant property, when we could simply add 1 more dimension if we wished to expand the capacity of the encoder; in comparison, the contribution of this one additional norm value seems insignificant. Yet it seems the norm does play at least a subtle role in encoding information, in spite of the fact that none of the contemporary encoding algorithms are explicitly designed to store information there. We have shown that a signal can be stored there, and the way it is used as a storage container seems to depend on an interaction between the encoder architecture and the type of information that is being encoded. This is a valuable interpretability finding, regardless of whether it has further applicability.

Regarding the question of applicability, aside from (a) the method offering valuable interpretability insights and (b) its aforementioned impact on reframing existing and future research, (c) the finding that the norm is an information container could have relevant applications on model design in the future. There is certainly potential to leverage the knowledge that embeddings have this seemingly inherent capability of encoding information not just in dimensions, but also in the norm. We can envisage an application where new encoders can be designed, or existing encoders modified, to explicitly store suitable information in the norm container which best corresponds to the linear nature of the norm. This might benefit embeddings in the sense that it would free up representation space in the dimension container, in turn making the encodings more streamlined and efficient.

8.1 Limitations

The research presented in this thesis has yielded many insights into how different types of linguistic information are encoded in embeddings. While valuable on their own merit, our findings also serve the purpose of validating the newly proposed *probing with noise* method,

demonstrating that it can produce relevant insights and can generalise to a number of different types of embeddings, encoders and probing tasks. To this end, we have cast a wide net and favoured a broad approach rather than diving too deep into any of the topics presented in the thesis. All the topics covered here can be pursued in more depth and the research can be taken further, and as such our work comes with certain limitations, which we acknowledge and address here.

It is always possible to expand and extend any line of work, and while it is true that "we could have done more experiments" is not a valid limitation or criticism, it does evoke an inherent limitation of our research, which suffers from the general limitations of any empirical work: the work in this thesis measures behaviours on a large number of data points and attempts to draw conclusions from these measurements. With this always comes the risk that our conclusions hold only for the datasets on which we measured or the models which were used to measure, be it embeddings, probes or probing tasks. While our research scope has been quite broad, encompassing examples of taxonomic, thematic, contextual and static embeddings, as well as probing for different linguistic domains, there is still a distinct possibility that our findings might not generalise to other settings. While this issue is more epistemological in nature than it is specific to our work, it is still worth considering what other avenues could have been explored, if for no other reason than to inspire new directions for future work.

Encoders We have only explored some of the historically most popular language encoders. While we purposefully chose embeddings that represent encodings of different types of information (i.e. taxonomic vs thematic, contextual vs. static), to truly be able to draw general conclusions about the way any of these types of embeddings encode information, a much more comprehensive study would be needed with a sole focus on each encoding type.

The same criticism can be applied to our choice of taxonomic embeddings, which was anything but trivial. Our experiments examined only one type of taxonomic embeddings—

ones trained on a pseudo-corpus generated from a WordNet random walk. A number of variables could have been different in this scenario: we could have chosen a different algorithm than the random walk, we could have applied the random walk to a different underlying taxonomy, we could have used a different encoding model other than SGNS or GloVe, or we could have used any of the other pseudo-corpora that were generated in Chapter 4 using different hyperparameters, or any combination of the above. Each one of these variables could have an impact on the experimental results, and all of them are options that exist in addition to the alternative approach of foregoing the random walk algorithm completely and instead examining other types of taxonomic embeddings.

As a fortunate consequence of our decision to use only one type of taxonomic embedding algorithm, our SGNS and GloVe taxonomic embeddings were both trained on the same pseudo-corpus, meaning that we have controlled for the training data, which gives us confidence that the observed differences are a product of the different encoder architectures. This, however, cannot be said for our usage of thematic off-the-shelf SGNS, GloVe and BERT embeddings. A limitation that arises when considering the performance of our thematic embeddings, which has in part been already discussed in Section 5.6, is that they have been trained on completely different datasets of dramatically varying sizes and content. To truly test the impact of their architectures on the probing tasks, the training data based upon which their word embeddings are generated should be identical between all three encoders. Certainly, implementing this was not feasible in practice, and using off-the-shelf varieties provided insight into the functioning of well-known and commonly used embeddings, but it consequently limits the comparability of their results as we cannot confidently distinguish whether differences in performance are due to differences in architecture or training data.

Another source of uncertainty stems from the way we generate the sentence embeddings needed to probe for sentence-level information. All the encoders we have used, be it taxonomic, thematic, contextual or static, generate word-level embeddings. In Chapters 6 and

7 we opted for averaging the word embeddings in each sentence, also known as mean pooling, as this is one of the most popular ways to generate sentence representations. However, there are other known approaches available to choose from, such as max pooling and min pooling, where we extract the most salient features from every word embedding dimension by taking the maximum or minimum value along each dimension of the word vectors in the sentence to generate a sentence representation (Shen et al., 2018). When it comes to BERT, additional options are available, such as taking the CLS token representation, which is BERT's own sentence representation. Finally, in BERT we averaged the representations from the final layer, but we could have taken embeddings obtained from other layers as well.

Probes Throughout our host of experiments, we have consistently used only one probing classifier, an off-the-shelf MLP implementation using its default parameters. This was done consciously, in order to avoid adding another variable to our experimentation and thereby increasing the complexity of our experiments. However, as already pointed out in Section 3.1, the probe used for our method needs to be able to take a global view of the input features in order to have access to the norm container. When applying our method, we need to remain conscious that not all probes will be able to distinguish between the two information containers. This might be especially important in light of the ongoing discussion about information extractability in the literature as presented in Chapter 2, which aims to judge a probe's ability to extract information from an encoding. As we have not tested whether our method is able to provide a stronger or weaker signal using other kinds of probing classifiers, or whether it would provide a signal at all, we cannot claim that our method generalises to other probes. However, even in the case where the probe cannot inherently access the norm container, this can be worked around by simply adding the norm value to the vector explicitly, before giving it to the probe for training. In any case, a relevant avenue of research would be to explore different probing classifiers, or to design different probes altogether.

Tasks In our work we have sampled a wide array of probing tasks that represent a number of language domains, in an attempt to identify the types of linguistic information that can be encoded in the norm. However, the same criticism that applies to our choice of encoders can be applied to our chosen set of probing tasks: for our results to truly be representative of any given language domain, we would need a comprehensive study with a sole focus on syntax, semantics, morphology etc. But even beyond such deep dives into the distinct language domains, the probing tasks that we have introduced ourselves also come with certain limitations, which we have in part discussed previously, but reiterate here for completeness.

The hypernym-hyponym probing task introduced in Chapter 5 was constructed to represent the underlying taxonomy and the relations between hypernym-hyponym pairs. However, WordNet's vocabulary, on which the probing dataset is based, is indeed quite small, and an MLP is a powerful probe with the capability to memorise data points. Even though we have gone to some lengths to avoid possible confounders and the risk of lexical memorisation, given the generally high performance of the vanilla classifiers we wonder whether there is still a chance that the probe might have simply memorised the individual word embeddings, rather than learning the hypernym-hyponym relations between word pairs. There is not much more we could have done, aside from hand-picking the candidates and making sure that a given lemma never appears in both the train and test set to avoid memorisation. However, even if we had done this, it is not clear whether that would have helped or created an unnecessarily biased dataset: given that the probe's inputs are made up of hypernym-hyponym pairs, many lemmas will be co-appearing in both the hypernym and hyponym role. Cleanly separating the dataset using this criterion might create a severely skewed train and test sample, where a balanced split might not even be impossible to achieve.

Meanwhile, the limitations of the idiomatic usage probing task have been discussed at length in Section 6.5. When taken together with the post hoc analysis, the findings from this dataset seem confusing and inconsistent with the findings observed on other datasets. Even if

the results were not inconclusive, they might not generalise to other idiomatic expressions that are not verb-noun combinations. Of course, it is easy to say in hindsight that we should have chosen a more suitable dataset. However, at the conception of this work not as many datasets were freely available—several of the larger datasets mentioned in Chapter 6 have been released in the past 2 years, when our work was already underway—and we chose this particular subset of idiomatic expressions because we wanted to work on an existing dataset already used in the literature so that we had previous work to compare against. Yet there exist many other types of verbal multi-word expressions, let alone non-verbal idiomatic phrases. A more exhaustive dataset would have to be curated for a more thorough and general analysis of idiomaticity as such, rather than just idiomaticity of VNCs.

Post hoc analysis When it comes to our *probing with noise* method, it offers a clear starting point from which further expanded and more targeted post hoc experiments can be done. We have exemplified this with dimension deletion experiments and a Pearson correlation study, which was an obvious first choice for the post hoc analysis of the norm. However, it is important to be aware that the Pearson test comes with the limitation of only describing linear relationships, whereas it is possible that connections between variables can be non-linear. Fortunately we have not encountered instances where this would be the case in any of our experiments (i.e. a scenario where the MLP probe detects a signal in the norm, but the Pearson correlation on the same task is $\approx$0), but we do acknowledge that more appropriate statistical tests can be performed. While we have shown that even this limited correlation test, as well as a coarse-grained dimension deletion experiment can provide valuable insights, so much more can be done to study both the norm and the dimension container, and here we have just barely scratched the surface.

Finally, we stress that we stand by all the choices we have made, as they have all been made in a sound, informed and methodologically consistent manner, and all the resulting experiments have significantly contributed to the thesis as a whole. However, we did wish

to highlight just how many choices have been made along the way, and how the number of alternative paths grows exponentially the further back up the decision tree we look. While there is nothing fundamentally wrong about the work that has been carried out, each choice could have made for a drastically different suite of experiments and could potentially have yielded different results. In fact, we find this to be a very strong and exciting motivator for future work, as this long list of "missed opportunities" only goes to show how young and rich this research area still is and how many more avenues there are to explore, with new insights waiting to be uncovered.

8.2 Future Work

While the research presented in this thesis has provided many insights into how different types of linguistic information are encoded in embeddings, certain questions still remain open. The high modularity of our *probing with noise* method and its proven applicability to a wide array of different probing tasks suggests a high likelihood that swapping in other models and tasks would produce valid results and provide the types of insights we have presented in the thesis. Rather than lamenting what could have been, we take inspiration from the limitations section and consider a number of potentially fruitful avenues that can be pursued in future work, suggesting a series of experiments that will test how our method applies to a number of different permutations of its pipeline.

1. A host of studies focused on embedding algorithms As stated in the previous section, a comprehensive study with a sole focus on a given encoding type is needed to make general conclusions about how a certain type of embedding encodes information. Given their current prominence, a study of contextual encoders would be an appropriate starting point, for example comparing different architectures such as BERT, ELMo and XLNet. Another pertinent research direction is to be even more fine-grained and run a study comparing a

number of BERT's direct derivatives like ALBERT (Lan et al., 2019), BART (Lewis et al., 2020), DistilBERT (Sanh et al., 2019) and RoBERTa (Liu et al., 2019b), among others, thus contributing novel insights to the growing field of BERTology (Rogers et al., 2020).

On the other hand, we would also be keen to study other types of taxonomic embeddings that are not based on a WordNet random walk (such as Poincaré embeddings (Nickel and Kiela, 2017), Embedding of Semantic Predications (Cohen and Widdows, 2017) or Personalised PageRank-based algorithms (Agirre et al., 2010)). Measuring their performance on our hypernym-hyponym probing task would provide solid insight into how taxonomic information can be encoded in embeddings. Furthermore, while we do not expect groundbreaking results, we are keen to complete the missing puzzle piece and apply our method to taxonomic embeddings on the non-taxonomic linguistic probing task datasets introduced by Conneau et al. (2018), to see whether any non-taxonomic sentence-level information is present in their norm, or indeed anywhere in the vectors at all.

On a related note, we are also interested in investigating the performance of thematic embeddings on word-level probing tasks, as well as devising a sentence-level taxonomic task on which we can evaluate our taxonomic embeddings. This brings us to the next general line of work we would like to see studied in the future: expanding the application of our method to a wider variety of tasks.

2. A host of studies focused on other semantic probing tasks Given that the question whether idiomaticity can be encoded in the norm remains open, we are quite keen to find an answer. To begin improving the work in this space, we propose starting with updating the VNC-tokens dataset for idiomatic usage, which has proven to be a somewhat underwhelming resource for idiomaticity probing. What is needed is a deep review and cleaning of the existing annotations, aligning the dataset with the PARSEME annotation guidelines and sourcing additional examples of sentences containing idiomatic and literal examples of the VNCs in the dataset, with the aim of improving the balance of idiomaticity labels. If at all

possible, it would be wise to also attempt to control for sentence length, as this could be a confounding factor. This line of work would certainly improve the quality of the dataset, which could be released as version 2.0, specially curated to be a probing task dataset.

Furthermore, we would be very interested in widening the scope of idiomatic expressions that are studied in the probing literature. To this end, we can do two things: (a) create an amalgam of all existing idiom datasets in order to increase training size and apply our method to probe for a very general encoding of idiomaticity, or (b) apply our method to different datasets individually in order to see whether there are any regularities or perhaps differences in the ways different kinds of idiomatic phrases are encoded in vector space. To take this a step further, we would be interested to expand the use and availability of semantic probing tasks beyond idiomaticity or taxonomic information. Avenues are plentiful, while more interesting ones include tasks like metaphor prediction, polysemy detection and word association datasets.

It is also worth considering that most encoders are trained on standardised corpora, often web content and news text (e.g. the Google News dataset used to train the word2vec model described in Section 5.3), where the frequency of idiomatic language use is relatively low. However, many natural language texts come from the domain of fiction, where the literary language is highly poetic, idiomatic and often allegorical. In such texts, the frequency of idiomatic language use is significantly higher, yet most dataset and models do not explicitly account for this. Thus another research avenue presents itself in training embeddings on literary texts and probing them on a multitude of idiomatic usage and metaphor prediction datasets, comparing their behaviour and performance to embeddings trained on more standard corpora.

Finally, as part of our efforts to include other datasets, we consider the merits of adding another dimension to our line of work by adopting a cross-lingual perspective. So far we have only applied our method to English datasets, yet it would be extremely informative and

beneficial to study its application to probing datasets in other, more typologically diverse languages (Bender, 2019). This includes using existing multilingual embeddings and probing datasets, as well as developing and publishing new ones. The latter would especially benefit the NLP landscape of low-resourced languages like Croatian and Irish, which we take personal interest in. Findings based on cross-lingual comparisons would certainly bring more complexity to the table, but would also result in valuable and more nuanced insights.

3. Additional post hoc analyses There are further post hoc analyses we can run based on the datasets we have presented in this thesis, mainly focused on identifying where in the dimension container the relevant information is encoded, with an aim of being more precise and less coarse-grained than our dimension deletion experiments have shown to be. We propose a series of post hoc analyses in order to achieve this: by considering vector dimensions as being feature vectors, we can perform statistical analyses typical for standard machine learning pipelines to check whether certain dimensions are correlated.

This includes tests such as collinearity analysis which can help determine correlations between individual dimensions. We can also perform clustering over the feature vectors, in an attempt to identify which dimensions correspond to the class labels and whether there might be any outliers (particularly relevant for the dubious idiomatic usage dataset). We can also quantify such differences by calculating pairwise cosine similarity scores[6] for instances in our dataset to measure the similarities between idiomatic and literal instances. Finally, we can apply dimensionality reduction techniques and principal component analysis to help identify relevant dimensions, and we can train a new probing classifier on the resulting representations in order to examine how such changes impact the probe's performance. Such exploratory approaches would allow us to more precisely identify where in the embedding idiomaticity is encoded and how it affects the various aspects of the feature vectors.

[6] As here we would be interested only in information encoded in the dimension container, a cosine similarity calculation is appropriate. If we wished not to lose the information encoded in the norm, then we would need to forego the normalisation step and just calculate the dot product.

We note that this line of work is parallel to the research of Torroba Hennigen et al. (2020) and Durrani et al. (2020), among others, and we suspect that adopting their approaches and applying it to our datasets would also yield relevant insights. Indeed, this would make for an interesting replication experiment, and a comparison of results obtained in this replication study with the statistical analyses described in the previous paragraph would help validate the findings.

Furthermore, we can design *probing with noise* scenarios that are specialised for diagnosing linguistic confounding factors. Framing a subsequent iteration of *probing with noise* experiments as a post hoc analysis could help us determine, for example, whether information encoded in the norm of idiomatic and literal sentence embeddings indeed corresponds to idiomaticity, or perhaps some other linguistic signal. To obtain this insight, we can annotate sentences from the IU dataset for some other linguistic property and use the sentences with idiomatic phrases as training data for this other linguistic probing task. As an example, the simplest one to execute would be sentence length—we can automatically attach sentence length annotations to each sentence in the VNC tokens dataset and then use the IU sentence embeddings to predict their length. Then we can see if any of them reveal that sentence length information is encoded in the norm of the vectors. If found to be true, this might help us identify that particular type of linguistic information as a confounder of idiomatic usage information. With some additional annotation work, we could perform the same study for any of the other linguistic categories. While this is mainly useful for the idiomatic usage and hypernym-hyponym datasets, as the datasets published by Conneau et al. are decorrelated and have accounted for most confounders, conceptually it is a useful tool to keep in our arsenal.

4. Identifying the linguistic signal Finally, we would like to pursue the line of work we have set up earlier in this chapter: after running further idiomaticity experiments, regardless of whether idiomaticity turns out to be encoded in the norm or not, what we can already

state with certainty is that our experiments and related literature indicate that embeddings do contain some notion of idiomaticity, however imperfect (Garcia et al., 2021). On a more abstract level, we would be interested to use our framework to identify the linguistic signal that the encoders use to model this semantic phenomenon.

We plan to implement a series of experiments that will help us identify which linguistic signal is contained in the input sentences that is being encoded in the sentence embeddings and in turn picked up by the probe to classify sentences with idiomatic or literal usage. In staying true to our method, we would approach this issue by introducing noise into the pipeline. Rather than introducing it into the embeddings, we will introduce noise into the dataset.

Based on the assumption that there is a quantifiable linguistic signal present in a sentence containing an idiomatic phrase that indicates whether there is idiomatic usage or not, we hypothesise on what that signal might be, and then modify the sentence to introduce noise in such a way as to disrupt that signal. We would train sentence embeddings and probe them on the same task to see whether their performance drops. If it does, this means we have identified a type of linguistic noise that interferes with the model's encoding of idiomaticity, in turn identifying what the signal actually is. While our interests are in the space of semantics, so naturally we choose a semantic task for this, conceptually this approach can be applied to any type of linguistic information, all it requires is a good set of adversarial, disruptive interventions in the datasets.

Similar work in this direction has already been done: Nedumpozhimana and Kelleher (2021) run a set of masking experiments on BERT, testing the assumption that the surface form of idiomatic phrases is the signal for predicting idiomaticity—by masking the surface forms, they remove this signal, and use the results to analyse where in a sentence idiomatic information is taken from. Similarly, we offer another noise candidate: artificial perplexity. Our experiment would be based on the assumption that, rather than surface forms, contextual

incongruity[7] is the relevant signal for encoding idiomaticity. If we replace idiomatic phrases with highly infrequent non-idiomatic words that are simply out-of-context, this could impact contextual incongruity in a similar fashion an idiomatic phrase does. If introducing such noise does not have an effect on the performance of the probe, then that confirms that the embedding models can pick up on contextual incongruity and directly encode that in the representations.

[7] Assuming that incongruity can be measured as high perplexity, hence the name "artificial perplexity".

Chapter 9

Conclusion

We have stated a number of research questions in the introduction to the thesis, as our aim during the course of the PhD was to address the following issues:

- Q1: How are different types of linguistic information encoded in embeddings?
- Q2: Is the vector norm of embeddings capable of encoding certain linguistic properties?
- Q3: What is the interaction between different types of embeddings and the way they encode linguistic properties?

Notably there is significant overlap between the three questions and many of our individual efforts to answer them often address more than one issue at a time. Hence, in order to provide the insights necessary to answer these questions, we have made a number of compounding research contributions that fall on the intersection of three fields of study—semantics, embeddings and probing.

One of the main contributions of this thesis is the development of a methodological extension of the probing framework which we call *probing with noise*. An extensive experimental evaluation provides evidence that supports the viability of the method, showing that it can generalise to a number of different types of embeddings, encoder architectures and probing

tasks. The method reveals the existence of separate *information containers* in embeddings at both the word and sentence level, demonstrating that linguistic information encoded between the dimensions and the norm can be redundant, but also supplementary, and, most strikingly, that the norm is able to contain information that the dimensions do not. We also show that the method can act as a kind of presupposition test for any structural investigation of embeddings, as it provides insight into where in the embeddings certain linguistic information is contained. Once this is established, it can facilitate a number of post hoc experiments and analyses can be performed to better understand the nature of information encoded in embeddings.

The development of the method thus answers Q1 as it helps us understand how different types of linguistic information are encoded in embeddings. In showing that the norm is able to contain information that the dimensions do not, our work answers Q2. The method also offers a framing of targeted structural analyses as post hoc experiments, which allow for a better understanding of the nature of information encoded in embeddings, thus opening the door towards addressing Q3, showing that different encoders use the norm to store different amounts of information, as well as that a certain amount of redundancy exists between norm and dimensions, as well as within the dimensions themselves.

In addition to structural properties of embeddings, all three research questions are concerned with the types of linguistic information and linguistic properties that embeddings can encode. We have made a number of contributions that illuminate these issues by studying the taxonomic and thematic dimensions of semantic information. To facilitate this comparison we have trained taxonomic word embeddings that are trained on WordNet random walk pseudo-corpora. This has allowed us to expand our understanding of the random walk algorithm and the relationship between the structure of the underlying knowledge graph, the properties of the pseudo-corpora generated from the graph, and the performance of the embeddings trained on these pseduo-corpora, showing that some pseudo-corpora derived

from WordNet's taxonomy resemble natural corpora at a statistical level. In addition, the pseudo-corpora and embeddings have also been made publicly available.

We also used our *probing with noise* method to study the differences between our taxonomic embeddings and off-the-shelf thematic embeddings, and for this purpose developed a new semantic probing task for hypernym-hyponym prediction. Applying our method on this dataset has shown that, while the majority of the relevant information is encoded in the dimensions of both taxonomic and thematic embeddings, only taxonomic embeddings carry the information pertinent to the hypernym-hyponym task in their norm, indicating that the role of the norm can be determined by the embedding training data, rather than the embedding model architecture. In terms of additional structural insights into embeddings, we have found that when it comes to the vector space of our taxonomic embeddings, hypernyms are positioned further away from the origin of the space than hyponyms are.

In order to also study differences between two different types of thematic embeddings—contextual and static—we repurposed an existing idiom dataset for a probing task of idiomatic usage prediction to be used as a thematic semantic probing task. We have found that a probe trained on a contextual encoder is better than a static encoder at predicting the task. However, our method indicates that idiomaticity is not encoded in either of the encoders' norms, in spite of the fact that a post hoc analysis shows that the norm of sentences containing idiomatic usage is shorter, meaning they are located closer to the origin of the space relative to sentences with literal usage. Compounding these inconsistent results, we have observed a number of inconsistent and unexpected behaviours on this task that indicate that the dataset itself has some limitations that might affect our probe's performance. However, in repurposing the dataset to be used as a probing task, we have established a number of strong guidelines as to what properties a general idiomatic usage probing task train and test split should reflect, which can inform future work on the topic.

Finally, we have applied our method to ten additional tasks that represent a wider selection of language domains beyond semantics, such as surface information, morphology, syntax and contextual incongruity. We have found that on most language tasks the vast majority of the relevant information is encoded in the dimension values, but can sometimes be supplemented with information from the norm. Which type of information is encoded in such a way seems to be dependent on the encoder, as different encoders have been shown to store different types of information in the different containers. We have thus learned that, true to its name, a contextual encoder mainly encodes contextual incongruity information in the norm, while a static encoder mainly uses it to store syntactic and surface level information. We have also learned about the ways these properties are encoded in the vector space: the deeper the syntactic parse tree, the further away the sentence is positioned from the origin by a static encoder, while in a contextual encoder sentences containing contextual incongruity are located closer to the origin.

In terms of differences between contextual and static encoders, in addition to identifying which types of information they encode in which container (i.e. syntactic vs contextual), we have also found that there are differences in how information tends to be localised in their respective dimension containers: both encoders exhibit a certain degree of information localisation in their dimension container, with a possible preference for certain dimensions to hold certain information, and there seems to be a high degree of information redundancy across the dimensions in a vector, not just between the dimensions and the norm. GloVe exhibits a blanket preference for the first half of the vector dimensions, and BERT shows localisation tendencies only for certain language tasks.

In conclusion, we have performed an exploratory, empirical study of the geometric properties of different types of embeddings. We have extended the existing probing framework and devised a method that allows us to peek deeper into the black box of language representations. By performing a systematic exploration of the importance of the vector norm in encoding

different types of linguistic phenomena in different embedding models, we have expanded our understanding of the structural properties of certain models and the way they encode information. While our findings contribute to insights relevant to the domain of model interpretability, an equally relevant takeaway is that our method provides the type of insight that can facilitate more principled approaches to structural research on embeddings. By identifying the information containers that are relevant for encoding the target information, the method allows us to explicitly test our presuppositions regarding the location of the relevant information in embeddings, thus making our method a necessary prerequisite step to doing structural analysis, and allowing us to make considerations about how a given vector modification might impact the information containers.

Probing with noise can provide new perspectives and broaden our understanding of embeddings. However, our work is by no means exhaustive: further, deeper and expanded applications of the method, such as exploring a host of other representations, different pooling strategies or tracking behaviour across embedding layers, exploring word-level tasks or folding in additional datasets, are all fruitful avenues for future work. Fortunately, the method is robust enough to be applied to any encoder and any dataset, whether it is at the word or sentence level, which will allow for streamlined and systematic further study. This type of analysis can lead us towards providing insight into the language signals that the encoders use to recognise the presence of linguistic phenomena, informed by the different types of information that different encoders store in their respective geometric components.

Appendix A

Pearson Correlation Analysis of L1 and L2 Normalised Embeddings

Table A.1 presents an extended Pearson correlation analysis that includes correlations between class labels and the norms of L1- and L2-normalised vectors, in addition to vanilla vectors and vectors with ablated norm information using our noising function as described in Section 3.3.2.

As supported by Goldberg (2017, page 117), the results show that normalising the vectors removes information encoded in the norm. This does seem to come with a caveat, though: normalisation only removes information from the same order norm as the normalisation algorithm. We can observe this in the table: applying an L1 normalisation algorithm to the vectors seems to completely remove any information encoded in the L1 norm, as the correlation drops to ≈ 0. The same happens to the correlation with the L2 norm when applying L2 normalisation. However, surprisingly, it seems that a given normalisation algorithm impacts the other norm as well. For example, in the BS task L2 normalisation nullifies the L2 norm's correlation with the class labels, but in turn strengthens that correlation for the L1 norm, which intensifies from -0.39 to -0.44. On the other hand, L1 normalisation

causes the same strengthening of correlation in the L2 norm, but also changes the sign—the L2 norm's correlation with BS class labels increases from -0.32 to 0.43.

Additionally, the L1 norm has a stronger correlation with the class labels than the L2 norm in all tasks except IU for BERT, and CI and TD for GLOVE, where the opposite is true. This shows that while both norms correlate with some class labels, the degree in which they do differs, indicating the information they encode is slightly different, and that there is incomplete overlap between what, or how much, the two norms encode.

This shows that on certain tasks, not only is the other norm unaffected by a normalisation procedure, but its correlation with the task labels increases. We observe this to varying degrees in SL, ON, TE and BS. Furthermore, while the correlation weakens in SOMO and IU, it still exhibits the latter behaviour—the sign changes when the vectors are L1 normalised, but not when they are L2 normalised. This is prevalent across all datasets, even in cases where the correlation between norm and class labels is $\approx$0.

This analysis supports our decision from Section 3.3.2 to use a different noising function to remove information from the norm container, as only the vectors with fully ablated norms have an $\approx$0 correlation with both the L1 and L2 norms.

Task	Vectors	**GloVe**		**BERT**	
		L1	L2	L1	L2
SL	Vanilla	-0.7278	-0.3758	-0.1564	-0.1039
	L1 Normalised	-0.0013	0.7161	0.0032	0.2195
	L2 Normalised	-0.7027	0.0001	-0.2223	0.0001
	Abl. norm	-0.1893	-0.0025	-0.0417	-0.0013
SN	Vanilla	0.0360	0.0268	0.0071	0.0146
	L1 Normalised	0.0028	-0.0228	-0.0010	0.0087
	L2 Normalised	0.0255	-0.0019	-0.0086	-0.0003
	Abl. norm	0.0036	-0.0033	-0.0035	-0.0021
ON	Vanilla	0.0013	0.0008	-0.0736	-0.0583
	L1 Normalised	-0.0016	0.0048	-0.0015	0.0892
	L2 Normalised	-0.0004	-0.0015	-0.0901	0.0037
	Abl. norm	0.0009	0.0013	-0.0181	-0.0010
TE	Vanilla	-0.1152	-0.0571	-0.0542	-0.0413
	L1 Normalised	-0.0020	0.1040	-0.0023	0.0659
	L2 Normalised	-0.1071	-0.0006	-0.0691	-0.0018
	Abl. norm	-0.0317	-0.0007	-0.0116	0.0010
TD	Vanilla	-0.0817	0.1908	-0.0415	-0.0251
	L1 Normalised	0.0005	0.3133	0.0021	0.0645
	L2 Normalised	-0.3159	-0.0026	-0.0652	0.0000
	Abl. norm	-0.0665	0.0016	-0.0163	-0.0045
CIN	Vanilla	-0.0019	-0.0094	-0.0755	-0.0638
	L1 Normalised	0.0000	-0.0062	-0.0047	0.0846
	L2 Normalised	0.0065	0.0064	-0.0850	0.0034
	Abl. norm	0.0029	0.0018	-0.0152	-0.0015
BS	Vanilla	0.0040	0.0002	-0.3866	-0.3238
	L1 Normalised	-0.0015	-0.0048	0.0004	0.4333
	L2 Normalised	0.0056	-0.0019	-0.4357	0.0024
	Abl. norm	0.0022	0.0006	-0.0978	-0.0005
SO MO	Vanilla	-0.0464	-0.0222	-0.2414	-0.2305
	L1 Normalised	0.0031	0.0401	0.0035	0.2213
	L2 Normalised	-0.0392	-0.0014	-0.2219	0.0023
	Abl. norm	-0.0105	0.0000	-0.0420	0.0021
IU	Vanilla	-0.2231	-0.1786	-0.1490	-0.1756
	L1 Normalised	-0.0019	0.1540	-0.0241	0.0932
	L2 Normalised	-0.1317	0.0137	-0.0924	0.0125
	Abl. norm	-0.0074	0.0276	-0.0397	-0.0167

Table A.1 Pearson correlation coefficients between the class labels and vector norms for vanilla vectors, L1 and L2 normalised vectors, as well as vectors with ablated L2 norm containers.

Bibliography

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR, 2017*.

Eneko Agirre, Montse Cuadros, German Rigau, and Aitor Soroa. 2010. Exploring knowledge bases for similarity. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'10)*.

Wasi Uddin Ahmad, Xueying Bai, Zhechao Huang, Chao Jiang, Nanyun Peng, and Kai-Wei Chang. 2018. Multi-task learning for universal sentence embeddings: A thorough evaluation using transfer and auxiliary tasks.

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for nlp: A report on the first blackboxnlp workshop. *Natural Language Engineering*, 25(4):543–557.

Eduardo G. Altmann and Martin Gerlach. 2016. *Statistical Laws in Linguistics*. Springer International Publishing, Cham.

Howard Anton and Chris Rorres. 2013. *Elementary linear algebra: applications version*. John Wiley & Sons.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. *Handbook of natural language processing*, 2:267–292.

Kaspars Balodis and Daiga Deksne. 2018. Intent detection system based on word embeddings. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 25–35. Springer.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France. Association for Computational Linguistics.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, MD.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK. Association for Computational Linguistics.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Emily M. Bender. 2019. The #benderrule: On naming the languages we study and why it matters. *The Gradient*.

Emily M. Bender and Alex Lascarides. 2019. *Linguistic fundamentals for natural language processing ii: 100 essentials from semantics and pragmatics*, volume 12. Morgan & Claypool Publishers.

Yoshua Bengio. 2008. Neural net language models. *Scholarpedia*, 3(1):3881.

Gabriel Bernier-Colborne and Caroline Barrière. 2018. CRIM at SemEval-2018 task 9: A hybrid approach to hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 725–731, New Orleans, Louisiana. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Guido Boella and Luigi Di Caro. 2013. Extracting definitions and hypernym relations relying on syntactic dependencies and support vector machines. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 532–537, Sofia, Bulgaria. Association for Computational Linguistics.

Gemma Boleda, Abhijeet Gupta, and Sebastian Padó. 2017. Instances and concepts in distributional space. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 79–85, Valencia, Spain. Association for Computational Linguistics.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. SemEval-2015 task 17: Taxonomy extraction evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado. Association for Computational Linguistics.

Jordan Boyd-Graber, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. 2006. Adding dense, weighted connections to wordnet. In *Proceedings of the third international WordNet conference*. Citeseer.

Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana. Association for Computational Linguistics.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, Vancouver.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, and Kevin Gimpel. 2019. Evaluation benchmarks and learning criteria for discourse-aware sentence representations. 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 649–662.

Wanying Chiu and Kun Lu. 2015. Paradigmatic relations and syntagmatic relations: How are they related? *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Yejin Cho, Juan Diego Rodriguez, Yifan Gao, and Katrin Erk. 2020. Leveraging wordnet paths for neural hypernym prediction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3007–3018.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119, Athens, Greece. Association for Computational Linguistics.

Trevor Cohen and Dominic Widdows. 2017. Embedding of semantic predications. *Journal of Biomedical Informatics*, 68:150–166.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The vnc-tokens dataset. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 19–22.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. 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 4496–4505, Hong Kong, China. Association for Computational Linguistics.

Ferdinand De Saussure. 2011. *Course in general linguistics*. Columbia University Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Werner Ebeling and Thorsten Pöschel. 1994. Entropy and long-range correlations in literary english. *EPL (Europhysics Letters)*, 26(4):241.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. When bert forgets how to POS: Amnesic probing of linguistic properties and MLM predictions. *arXiv preprint arXiv:2006.00995*.

Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 424–435, Austin, Texas. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1606–1615, Denver, CO.

Manaal Faruqui and Chris Dyer. 2015. Non-distributional Word Vector Representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 464–469, Beijing.

Afsanesh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. In *Computational Linguistics*, volume 35, pages 61–103.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2020. CausaLM: Causal model explanation through counterfactual language models. *arXiv preprint arXiv:2005.13407*.

Anna Feldman and Jing Peng. 2013. Automatic detection of idiomatic clauses. In *Computational Linguistics and Intelligent Text Processing*, pages 435–446, Berlin, Heidelberg. Springer Berlin Heidelberg.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Lorenzo Ferrone and Fabio Massimo Zanzotto. 2020. Symbolic, distributed, and distributional representations for natural language processing in the era of deep learning: A survey. *Frontiers in Robotics and AI*, 6.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

John R. Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. 2014. Two is bigger (and better) than one: the Wikipedia bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 945–955, Baltimore, Maryland. Association for Computational Linguistics.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. 2016. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artificial Intelligence*, 241:66–102.

Winthrop Nelson Francis. 1964. A standard sample of present-day english for use with digital computers.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, Maryland. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Martin Gerlach and Eduardo Altmann. 2014. Scaling laws and fluctuations in the statistics of word frequencies. *New Journal of Physics*, 16:113010.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248.

Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. 2016. Single or multiple? combining word representations independently learned from text and wordnet. In *AAAI*.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random Walks and Neural Network Language Models on Knowledge Bases. In *Human Language Technologies: The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1434–1439, Denver, CO.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):117.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. 2019. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*.

Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Paşca, and Daniele Pighin. 2016. Revisiting taxonomy induction over Wikipedia. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2300–2309, Osaka, Japan. The COLING 2016 Organizing Committee.

Chongomweru Halimu, Asem Kasem, and S. H. Shah Newaz. 2019. Empirical comparison of area under roc curve (auc) and mathew correlation coefficient (mcc) for evaluating machine learning algorithms on imbalanced datasets for binary classification. In *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, ICMLSC 2019, page 1–6, New York, NY, USA. Association for Computing Machinery.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Chikara Hashimoto and Daisuke Kawahara. 2008. Construction of an idiom corpus and its application to idiom identification based on wsd incorporating idiom-specific features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 992–1001.

Chikara Hashimoto and Daisuke Kawahara. 2009. Compilation of an idiom example database for supervised idiom identification. *Language Resources and Evaluation*, 43(4):355–384.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Jim Hefferon. 2018. *Linear Algebra*. openintro.org.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv: Learning*.

Evan Hernandez and Jacob Andreas. 2021. The low-dimensional linear geometry of contextualized word representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 82–93, Online. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695.

Johannes Hoffart, Dragan Milchevski, and Gerhard Weikum. 2014. Stics: searching with strings, things, and cats. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1247–1248.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

V. Ivan Sanchez Carmona and Sebastian Riedel. 2017. How well can we predict hypernyms from word embeddings? a dataset-centric analysis. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 2, pages 401–407. Association for Computational Linguistics.

Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–65, Brussels, Belgium. Association for Computational Linguistics.

Magdalena Kacmajor and John D. Kelleher. 2019. Capturing and Measuring Thematic Relatedness. *Language Resources and Evaluation*, pages 1–38.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Jussi Karlgren and Pentti Kanerva. 2021. Semantics in high-dimensional space. *Frontiers in Artificial Intelligence*, 4:123.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2015. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078.

Akihiko Kato, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Construction of large-scale English verbal multiword expression annotated corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

John D Kelleher. 2019. *Deep learning*. MIT press.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Milton King and Paul Cook. 2017. Supervised and unsupervised approaches to measuring usage similarity. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 47–52, Valencia, Spain. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28*, pages 3276–3284.

Filip Klubička, Lorena Kasunić, Danijel Blazsetin, and Petra Bago. 2022. Challenges of building domain-specific parallel corpora from public administration documents. In *LREC 2022 Workshop Language Resources and Evaluation Conference 25 June 2022*, page 50.

Filip Klubička, Alfredo Maldonado, and John D. Kelleher. 2019. Synthetic, yet natural: Properties of wordnet random walk corpora and the impact of rare words on embedding performance. In *Proceedings of GWC2019: 10th Global WordNet Conference*.

Filip Klubička, Alfredo Maldonado, Abhijit Mahalunkar, and John Kelleher. 2020. English WordNet random walk pseudo-corpora. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4893–4902, Marseille, France. European Language Resources Association.

Filip Klubička, Giancarlo D. Salton, and John D. Kelleher. 2018a. Is it worth it? budget-related evaluation metrics for model selection. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Filip Klubička, Antonio Toral, and Víctor M Sánchez-Cartagena. 2018b. Quantitative fine-grained human evaluation of machine translation systems: a case study on english to croatian. *Machine Translation*, 32(3):195–215.

Filip Klubička and Raquel Fernández. 2018. Examining a hate speech corpus for hate speech detection and popularity prediction. In *Proceedings of 4REAL: 1st Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language*.

Filip Klubička, Alfredo Maldonado, Abhijit Mahalunkar, and John D. Kelleher. 2020. English wordnet random walk pseudo-corpora. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4893–4902, Marseille, France. European Language Resources Association.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Arne Köhn. 2016. Evaluating embeddings using syntax-based classification tasks as a proxy for parser performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 67–71, Berlin, Germany. Association for Computational Linguistics.

Katarzyna Krasnowska-Kieraś and Alina Wróblewska. 2019. Empirical linguistic study of sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5729–5739, Florence, Italy. Association for Computational Linguistics.

Paul R. Kroeger. 2019. *Analyzing meaning*. Number 5 in Textbooks in Language Sciences. Language Science Press, Berlin.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Angeliki Lazaridou, Marco Marelli, and Marco Baroni. 2017. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive science*, 41:677–705.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada. Association for Computational Linguistics.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. 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, Denver, Colorado. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Linlin Li and Caroline Sporleder. 2010a. Linguistic cues for distinguishing literal and non-literal usages. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 683–691, Stroudsburg, PA, USA. Association for Computational Linguistics.

Linlin Li and Caroline Sporleder. 2010b. Using Gaussian mixture models to detect figurative language in context. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 297–300, Los Angeles, California. Association for Computational Linguistics.

Wei Li and Andrew McCallum. 2005. Semi-supervised sequence modeling with syntactic topic models. In *AAAI*, volume 5, pages 813–818.

Emilie L. Lin and Gregory L. Murphy. 2001. Thematic relations in adults' concepts. *Journal of experimental psychology: General*, 130(1):3.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside bert's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nikola Ljubešić, Željko Agić, Filip Klubicka, Vuk Batanović, and Tomaž Erjavec. 2018. hr500k–a reference training corpus of croatian. In *In Proceedings of the 2018 Language Technologies and Digital Humanities Conference (JT-DH 2018)*.

Alfredo Maldonado and Filip Klubička. 2018. ADAPT at SemEval-2018 task 9: Skip-gram word embeddings for unsupervised hypernym discovery in specialised corpora. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 924–927, New Orleans, Louisiana. Association for Computational Linguistics.

Alfredo Maldonado, Filip Klubička, and John D. Kelleher. 2019. Size matters: The impact of training size in taxonomically-enriched word embeddings. *Open Computer Science*.

Youness Mansar, Juyeon Kang, and Ismail El Maarouf. 2021. The finsim-2 2021 shared task: Learning semantic similarities for the financial domain. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 288–292, New York, NY, USA. Association for Computing Machinery.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 51–61.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12, Scottsdale, AZ.

Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Ilya Stutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the Twenty-Seventh Annual Conference on Neural Information Processing Systems (NIPS) In Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, NV.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden. Association for Computational Linguistics.

Vasudevan Nedumpozhimana and John D. Kelleher. 2021. Finding BERT's idiomatic key. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 57–62, Online. Association for Computational Linguistics.

Vasudevan Nedumpozhimana, Filip Klubička, and John D. Kelleher. 2022. Shapley idioms: Analysing bert sentence embeddings for general idiom token identification. *Frontiers in Artificial Intelligence*, 5.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Copenhagen, Denmark. Association for Computational Linguistics.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 454–459, Berlin.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6338–6347. Curran Associates, Inc., Long Beach, CA.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788. PMLR.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of NAACL-HLT*, pages 528–540.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ellie Pavlick and Marius Paşca. 2017. Identifying 1950s American jazz musicians: Fine-grained IsA extraction via modifier composition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2099–2109, Vancouver, Canada. Association for Computational Linguistics.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jing Peng and Anna Feldman. 2017. *Automatic Idiom Recognition with Word Embeddings*. Springer International Publishing, Cham.

Jing Peng, Anna Feldman, and Ekaterina Vylomova. 2014. Classifying idiomatic and literal expressions using topic models and intensity of emotions. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2019–2027, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018a. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018b. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2020. *Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning*, volume 13. Morgan & Claypool Publishers LLC.

Mohammad Taher Pilehvar and Nigel Collier. 2017. Inducing embeddings for rare and unseen words by leveraging lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 388–393.

Mohammad Taher Pilehvar, Dimitri Kartsaklis, Victor Prokhorov, and Nigel Collier. 2018. Card-660: Cambridge rare word dataset-a reliable benchmark for infrequent word representation models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1391–1401.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Yuval Pinter and Jacob Eisenstein. 2018. Predicting semantic relations using global graph properties. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1751, Brussels, Belgium. Association for Computational Linguistics.

Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Adversarial propagation and zero-shot cross-lingual transfer of word vector specialization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 282–293.

Sameer S Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. In *International Conference on Semantic Computing (ICSC 2007)*, pages 517–526. IEEE.

John Prager, Jennifer Chu-Carroll, Eric W Brown, and Krzysztof Czuba. 2008. Question answering by predictive annotation. In *Advances in Open Domain Question Answering*, pages 307–347. Springer.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Investigating language universal and specific properties in word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in bert. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102.

Marek Rei and Ted Briscoe. 2013. Parser lexicalisation through self-learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 391–400, Atlanta, Georgia. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

Erfan Sadeqi Azer, Daniel Khashabi, Ashish Sabharwal, and Dan Roth. 2020. Not all claims are created equal: Choosing the right statistical approach to assess hypotheses. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5715–5725, Online. Association for Computational Linguistics.

Ivan A. Sag, Thimothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Computational Linguistics and Intelligent Text Processing: Third International Conference: CICLing-2002, Lecture Notes in Computer Science*, volume 2276, pages 1–15.

Gözde Gül Şahin, Clara Vania, Ilia Kuznetsov, and Iryna Gurevych. 2020. LINSPECTOR: Multilingual probing tasks for word representations. *Computational Linguistics*, 46(2):335–385.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Giancarlo Salton, Robert Ross, and John Kelleher. 2016. Idiom token classification using sentential distributed semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 194–204, Berlin, Germany. Association for Computational Linguistics.

Giancarlo Salton, Robert Ross, and John Kelleher. 2017. Idiom type identification with smoothed lexical features and a maximum margin classifier. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 642–651, Varna, Bulgaria. INCOMA Ltd.

Giancarlo D. Salton, Robert J. Ross, and John D. Kelleher. 2014. An empirical study of the impact of idioms on phrase based statistical machine translation of English to Brazilian-Portuguese. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, pages 36–41, Gothenburg, Sweden. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Nine features in a random forest to learn taxonomical semantic relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4557–4564, Portorož, Slovenia. European Language Resources Association (ELRA).

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden. Association for Computational Linguistics.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8766–8774.

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1537–1547, Denver, Colorado. Association for Computational Linguistics.

Hinrich Schütze. 1993. Word space. In *Advances in neural information processing systems*, pages 895–902.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Melbourne, Australia. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 65–75, Valencia, Spain. Association for Computational Linguistics.

Kiril Simov, Petya Osenova, and Alexander Popov. 2017a. Comparison of word embeddings from different knowledge graphs. In *International Conference on Language, Data and Knowledge*, pages 213–221. Springer.

Kiril Simov, Alexander Popov, and Petya Osenova. 2015. Improving word sense disambiguation with linguistic knowledge from a sense annotated treebank. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 596–603.

Kiril Simov, Alexander Popov, and Petya Osenova. 2016a. The role of the wordnet relations in the knowledge-based word sense disambiguation task. In *Proceedings of Eighth Global WordNet Conference*, pages 391–398.

Kiril Ivanov Simov, Svetla Boytcheva, and Petya Osenova. 2017b. Towards lexical chains for knowledge-graph-based word embeddings. In *RANLP*, pages 679–685.

Kiril Ivanov Simov, Petya Osenova, and Alexander Popov. 2016b. Using context information for knowledge-based word sense disambiguation. In *AIMSA*.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, pages 1297–1304.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Pia Sommerauer, Antske Fokkens, and Piek Vossen. 2019. Towards interpretable, data-derived distributional semantic representations for reasoning: A dataset of properties and concepts. In *Proceedings of GWC2019: 10th Global WordNet Conference*.

Irena Spasić, Lowri Williams, and Andreas Buerki. 2017. Idiom-based features in sentiment analysis: Cutting the gordian knot. *IEEE Transactions on Affective Computing*, pages 1–1.

Robert Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679—-3686, Istanbul.

Robert Speer and Joanna Lowry-Duda. 2017. ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pages 85–89, Vancouver.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 754–762.

Felix Stahlberg, Danielle Saunders, and Bill Byrne. 2018. An operation sequence model for explainable neural machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 175–186, Brussels, Belgium. Association for Computational Linguistics.

Shuntaro Takahashi and Kumiko Tanaka-Ishii. 2019. Evaluating computational language models with scaling properties of natural language. *Computational Linguistics*, 45(3):481–513.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019*.

Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. 2020. Intrinsic probing through dimension selection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 197–216, Online. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Sara Veldhoen, Dieuwke Hupkes, and Willem H Zuidema. 2016. Diagnostic classifiers revealing how neural networks process hierarchical structure. In *Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (at NIPS)*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*.

Aline Villavicencio, Francis Bond, Anna Korhonen, and Diana McCarthy. 2005. Editorial: Introduction to the special issue on multiword expressions: Having a crack at a hard nut. *Comput. Speech Lang.*, 19(4):365–377.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.

Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Post-Specialisation: Retrofitting Vectors of Words Unseen in Lexical Resources. In *Proceedings of NAACL-HLT 2018*, pages 516–527, New Orleans, LA.

Yogarshi Vyas and Marine Carpuat. 2017. Detecting asymmetric semantic relations in context: A case-study on hypernymy detection. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 33–43, Vancouver, Canada. Association for Computational Linguistics.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany. Association for Computational Linguistics.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. 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 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Abigail Walsh, Claire Bonial, Kristina Geeraert, John P. McCrae, Nathan Schneider, and Clarissa Somers. 2018. Constructing an annotated corpus of verbal MWEs for English. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 193–200, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1203, Copenhagen, Denmark. Association for Computational Linguistics.

Andy Way, Petra Bago, Jane Dunne, Federico Gaspari, Andre Kåsen, Gauti Kristmannsson, Helen McHugh, Jon Arild Olsen, Dana Davis Sheridan, Páraic Sheridan, and John Tinsley. 2020. Progress of the PRINCIPLE project: Promoting MT for Croatian, Icelandic, Irish and Norwegian. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 465–466, Lisboa, Portugal. European Association for Machine Translation.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1015–1021, Geneva, Switzerland. COLING.

Henry M. Wellman and Susan A. Gelman. 1992. Cognitive development: Foundational theories of core domains. *Annual review of psychology*, 43(1):337–375.

Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors influencing the surprising instability of word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2092–2102, New Orleans, Louisiana. Association for Computational Linguistics.

Jess Whittlestone, Rune Nyrup, Anna Alexandrova, and Stephen Cave. 2019. The role and limits of principles in ai ethics: towards a focus on tensions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 195–200.

Dominic Widdows and Trevor Cohen. 2015. Reasoning with vectors: A continuous model for fast robust inference. *Logic Journal of the IGPL*, 23(2):141–173.

John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. 2015. From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Lowri Williams, Christian Bannister, Michael Arribas-Ayllon, Alun Preece, and Irena Spasić. 2015. The role of idioms in sentiment analysis. *Expert Systems with Applications*, 42(21):7375–7385.

Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, and Gerhard Weikum. 2013. Robust question answering over the web of linked data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1107–1116.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Zhiguo Yu, Trevor Cohen, Elmer V. Bernstam, Todd R. Johnson, and Byron C. Wallace. 2016. Retrofitting Word Vectors of MeSH Terms to Improve Semantic Similarity Measures. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 43–51, Austin, TX.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

George. K. Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human ecology*. Cambridge, Mass.: Addison Wesley Press, Inc.