Representation Learning for Word Senses and Evaluation of Their Properties

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Yi Zhou

May 2023

I would like to dedicate my work to my family and friends for their support, as well as to my three cats Jojo, Mia and Timo for their company.

Declaration

I hereby declare that, this thesis (and the work presented in it) is submitted to the Department of Computer Science, University of Liverpool, in fulfilment of the requirements for the degree of Doctor of Philosophy, and that this thesis is entirely my own work.

Liverpool, May 2023

Yi Zhou

Abstract

Contextualised word embeddings generated from Neural Language Models (NLMs), such as BERT and RoBERTa, represent a word with a vector that considers the semantics of the target word as well as its context. On the other hand, static word embeddings such as GloVe represent words by relatively low-dimensional, memory- and compute-efficient vectors but are not sensitive to the different senses of the word. To address the limitation of static word embeddings, we propose Context Derived Embeddings of Senses (CDES), a method that extracts sense related information from contextualised embeddings and injects it into static embeddings to create sense-specific static embeddings.

In addition to CDES, different methods have been proposed in prior work on sense embedding learning that uses different sense inventories, sense-tagged corpora and learning methods. However, not all existing sense embeddings cover all senses of ambiguous words equally well due to the discrepancies in their training resources. To address this problem, we propose a meta-sense embedding method – Neighbourhood Preserving Meta-Sense Embedding (NPMS), which learns meta-sense embeddings by combining multiple independently trained source sense embeddings such that the sense neighbourhoods computed from the source embeddings are preserved in the meta-embedding space. We show that our proposed method can combine source sense embeddings, which cover different sets of word senses.

Sense embedding learning methods learn different senses of ambigu-

ous words. One sense of an ambiguous word might be socially biased, while its other senses remain unbiased. In comparison to the numerous prior work evaluating the social biases in pretrained word embeddings, the biases in sense embeddings have been relatively understudied. We create a benchmark dataset, namely Sense-Sensitive Social Bias (SSSB), for evaluating the social biases in sense embeddings and propose novel sense-specific bias evaluation measures. We conduct an extensive evaluation of multiple static and contextualised sense embeddings for various types of social biases using the proposed measures. Our experimental results show that even in cases where no biases are found at the word level, there still exist worrying levels of social biases at the sense level, which are often ignored by the word-level bias evaluation measures.

Apart from the social biases, we evaluate the ℓ_2 norm of sense embeddings, which is another property of sense embeddings. We show that the ℓ_2 norm of a static sense embedding encodes information related to the frequency of that sense in the training corpus used to learn the sense embeddings. This finding can be seen as an extension of a previously known relationship for word embeddings to sense embeddings. Our experimental results show that, in spite of its simplicity, the ℓ_2 norm of sense embeddings is a surprisingly effective feature for several word sense related tasks such as (a) Most Frequent Sense (MFS) prediction, (b) Words in Context (WiC), and (c) Word Sense Disambiguation (WSD). In particular, by simply including the ℓ_2 norm of a sense embedding as a feature in a classifier, we show that we can improve WiC and WSD methods that use static sense embeddings.

Owing to all of the proposed methods in this thesis being monolingual, extending our methods and evaluations to cover multilingual sense embeddings is an important future direction. Moreover, using contextualised source embeddings as source embeddings to learn meta-sense embeddings, extending the categories of social biases in the SSSB dataset as well as developing debiasing methods for sense embeddings, improving dynamic word embeddings learning, and learning dynamic sense embeddings need to be further explored in future.

Acknowledgements

This thesis marks the end of a four-year journey at Liverpool, which is a very memorable and gratifying time of my life. I would like to thank everyone who support me and encouraged me during my PhD study. This thesis would not have been possible without their support.

I would like to begin by thanking my family who constantly support me to achieve my goal, none of this would be possible without them.

I would like to thank my supervisor Prof. Danushka Bollegala for supporting me in various aspects and providing feedback and guidance throughout my PhD. I learned a lot from him about scientific research methodologies. I have been privileged and pleased to have him as my supervisor. Without his guidance, this PhD. would not have been achievable.

I would like to thank Dr. Angrosh Mandya, Dr. Huda Hakami, Dr. Mohammad Alsuihaibani, Dr. Xia Cui, Dr. James O'Neill and Dr. Michael Abaho of the Natural Language Processing (NLP@Liv) Group at the Computer Science Department for all the helpful discussions we had throughout the years and the kind support and assistance whenever needed. I would also like to thank my collaborators Dr. Masahiro Kaneko, Guanqun Cao, Haochen Luo, Xiaohang Tang and Gaifan Zhang.

I thank my secondary supervisor Dr. Shan Luo and my IPAP member Prof. Xiaowei Huang and Prof. Martin Gairing for their valuable feedback and yearly review meetings.

Lastly, I owe thanks to the University of Liverpool Graduate Association (Hong Kong) and The Tung Foundation for providing the scholarship to support my study. I extend my thanks to the Department of Computer Science at the University of Liverpool, which equipped me with everything I needed to carry out my research.

Contents

List of Figures

List of Tables

Acronyms

NLP Natural Language Processing

WSD Word Sense Disambiguation

NLM Neural Language Model

LKB Lexical Knowledge Bases

AI Artificial Intelligence

CDES Context Derived Embeddings of Senses

WiC Words in Context

BPE Byte Pair Encoding

MFS Most Frequent Sense

SGNS skip-gram with negative sampling

CBOW Continuous Bag of Words

GLU Gated Linear Unit

LW Layer Weighting

PoS part-of-speech

MLM Masked Language Model

BPE Byte Pair Encoding

NPMS Neighbourhood Preserving Meta-Sense Embedding

SSSB Sense-Sensitive Social Bias

LSA Latent Semantic Analysis

SVD Singular Value Decomposition

IR information retrieval

HAL Hyperspace Analogue to Language

COALS Correlated Occurrence Analogue to Lexical Semantic

LR-MVL Low Rank Multi-View Learning

CCA Canonical Correlation Analysis

PIP Pairwise Inner Product

SoTA state-of-the-art

USM Uninformed Sense Matching

LSTM long short-term memory network

LM language model

RNN Recurrent Neural Network

CNN convolutional neural network

OOV out-of-vocabulary

WNG WordNet Gloss Corpus

KB Knowledge Base

ME Meta-embedding

1 Introduction

A word can possess one meaning (i.e., monosemous word) or multiple meanings (i.e., polysemous word). For instance, given two sentences:

- The grilled bass tastes good.
- He plays bass guitar.

The occurrences of the word bass denote two different meanings, which refers to *a type of fish* in the first sentence, while *a musical instrument* in the second. Each individual meaning of a word is regarded as its word sense. In spite of the breakthroughs in distributed semantic representations (i.e. word embeddings), addressing lexical ambiguity has remained a long-standing challenge in the field. Most of the time, humans have the capability to distinguish the correct meaning of a word unconsciously, whereas machines must analyse unstructured information in order to determine the correct meaning of an ambiguous word.

In the past decade, word embeddings that are learned by processing massive amounts of textual data via neural network based approaches have undoubtedly been one of the major points of attention in the Natural Language Processing (NLP) community. The introduction of static word embedding models, such as word2vec (Mikolov et al., 2013b), GloVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2017),

have generated a massive wave in the field of lexical semantics. Word embedding learning approaches map each word to a relatively low n-dimensional space, where two semantically or syntactically similar words become closer to each other.

Nevertheless, static word embeddings that combine information from various senses into the same representation, suffer from the limitation of being context insensitive and static, which refers to *meaning conflation deficiency* problem (Camacho-Collados and Pilehvar, 2018a), i.e., the inability to distinguish among various meanings of a word. As static word embedding learning models recast the same word type across various contexts, they are unable to distinguish different senses of polysemous words. In other words, static word embedding learning models use a single vector to represent each word in all contexts, regardless of the fact that a word may have different meanings in different contexts.

More recently, contextualised embeddings that are trained using Neural Language Models (NLMs), such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), etc., have shown the ability to learn contextualised representation. Contextualised embeddings are dynamic in the sense that the embedding of a word is learned depending on the context in which the word occurs. The effectiveness of NLMs has been proved on various NLP tasks, such as machine translation (Williams et al., 2018), grammatical error correction (Peters et al., 2017), speech recognition (Chiu et al., 2018), information retrieval (Conneau et al., 2017), summarisation (Howard and Ruder, 2018), question answering (Rajpurkar et al., 2016) and sentiment analysis (Dai and Le, 2015), to name a few. The main reason for this superior performance is their ability to capture the semantic and syntactic knowledge from contexts.

NLMs enable architectures to be built on top of them to achieve performances that were previously out of reach (Wang et al., 2019). Moreover, fine-tuning the same NLMs on numerous downstream tasks often results in comparable or even better performance compared with

sophisticated state-of-the-art (SoTA) task-specific models (Peters et al., 2019b). However, the latent representations learned from NLMs are not able to provide any information with regard to the word sense of the word in a given context. Such representations are not tied to a semantic network, such as WordNet (Miller, 1995) and BabelNet (Navigli and Ponzetto, 2012). Therefore, it is difficult to link them to structured sources of knowledge such as Lexical Knowledge Bases (LKB) (Scarlini et al., 2020b).

In order to tackle the aforementioned limitations in both static and contextualised word embeddings, numerous approaches have attempted to model individual meanings of words, i.e., word senses, as independent representations. Such representations are generally regarded as sense embeddings. Learning sense embeddings aims to build models to create robust multi-prototype semantic representations for different senses of words. The main idea for doing so is to augment the standard word embeddings by disambiguating word senses according to the contexts in which the words appear. Broadly, the field of sense embeddings can be categorised into two main paradigms, depending on how they learn distinct senses (Camacho-Collados and Pilehvar, 2018b):

- *Unsupervised*, where senses are learned directly from text corpora (Huang et al., 2012; Vu and Parker, 2016).
- *Knowledge-based*, where senses are associated with an external pre-defined sense inventory leveraging an underlying knowledge resource (Rothe and Schütze, 2015; Pilehvar and Collier, 2016; Mancini et al., 2017; Colla et al., 2020).

The work presented in this thesis aims to investigate and explore approaches to construct task-agnostic sense representations that can be used in multiple downstream tasks. The rest of this introductory chapter is organised as follows: Section 1.1 outlines the research aim and motivations for this thesis, Section 1.2 describes the research questions

and issues, Section 1.3 outlines the contribution of research, Section 1.4 lists publications that are peer-reviewed or currently under review and Section 1.5 describes the structure of the thesis and summarises this introductory chapter.

1.1 Research Aim and Motivation

The primary motivation for the work presented in this thesis is to develop approaches to learn sense embeddings, that enable machines to distinguish the correct word meanings for ambiguous words. The task of assigning the most suitable meaning to an ambiguous word in a given context is known as Word Sense Disambiguation (WSD). Given a word occurring in a context, WSD is the task of assigning the word with its most appropriate meaning selected from an external sense inventory. For instance, given the aforementioned sentence "The grilled *bass* tastes good.", the word *bass* must be associated with its *a type of fish* meaning according to a pre-defined sense inventory, such as WordNet.

WSD has been regarded as an Artificial Intelligence (AI)-complete problem (Mallery, 1988; Navigli, 2009; Camacho-Collados and Pilehvar, 2018b). Its challenges, which still exist at present, are multifold.

- Formalisation of the task depends on different sense embedding learning approaches, the granularity of sense inventories and domain of the corpus, etc;
- WSD heavily relies on knowledge sources.

However, the creation of such resources as well as the construction of sense-annotated corpora are time-consuming and require expensive effort. Furthermore, such sources and corpora need to be re-constructed for different domains and languages, as well as kept updated over time when the disambiguation scenario changes. This is an instance of the *knowledge acquisition bottleneck* (Gale et al., 1992).

In order to deal with the challenges in WSD, the work presented in this thesis aims to develop approaches to learn sense embeddings as well as investigate the properties of the learned sense embeddings. Below we summarise the specific research aims considered in the subsequent chapters of this thesis.

1. Chapter 3 - **Learning sense embeddings using contextualised embeddings as a proxy.** Static word embeddings represent each word by a single vector, regardless of the fact that a word may have different meanings based on different contexts. On the other hand, contextualised embeddings generated from NLMs represent a word with a vector that not only considers the semantics of the target word but also the context in which the word appears. Different types of information, such as word sense, dependency, and numeracy have been shown to be encoded in contextualised word embeddings. In addition, training large NLMs is expensive and time-consuming. Therefore, in this chapter, we aim to develop a lightweight method to learn static sense embeddings by extracting the sense information that is encoded in a pretrained NLM.

2. Chapter 4 - **There is no sense embedding covering all senses of ambiguous words equally well.** Existing sense embeddings are trained on diverse resources such as sense tagged corpora or dictionary glosses, with varying levels of sense coverage (e.g. fully covering all synsets in the WordNet vs. a subset), and using different methods. Therefore, the performance reported by the existing sense embeddings on different downstream tasks and datasets varies significantly for different part-of-speech (PoS) categories. In this chapter, we aim to address this problem by proposing a method to learn meta-sense embeddings. We combine multiple independently trained source sense embeddings such that the sense

neighbourhoods computed from the source embeddings are preserved in the meta-embedding space.

3. Chapter 5 - **Sense embeddings can be unfairly socially biased.** In comparison to the numerous prior work evaluating the social biases in pretrained word embeddings, the biases in sense embeddings have been relatively understudied. Therefore, in this chapter, we aim to evaluate the various types of social biases in both static and contextualised sense embeddings.

4. Chapter 6 - **The relationship between sense frequency and the ℓ_2 norm of sense embeddings.** Knowing the frequency of a sense helps to determine the majority sense, which has been used as a strong baseline for WSD. Prior studies (Arora et al., 2016; Mu and Viswanath, 2018) have shown that if word embeddings are anisotropic, the log-frequency of a word in a corpus is proportional to the squared ℓ_2 norm of the static word embedding, learned from the corpus. However, the relationship between sense embeddings and the frequency of a sense remains unclear. In this chapter, we aim to investigate the properties of sense embeddings by extending the prior results for word embeddings into sense embeddings.

1.2 Research Questions and Issues

In this section, we summarise the research questions that are addressed in this thesis.

1. In Chapter 3 we aim to study whether we can learn sense embeddings using the sense information encoded in contextualised embeddings. Specifically, we ask the following research question:

   Can we inject sense-related information extracted from contextualised word embeddings to create sense-specific

versions of (pretrained) static embeddings?

2. In Chapter 4 we aim to find out whether we can learn meta-sense embeddings by incorporating multiple independently trained source sense embeddings. Specifically, we ask the following research question:

   Can we learn sense embeddings that cover all senses of ambiguous words equally well, such that the sense-related information captured by the source sense embeddings is preserved in the meta-sense embedding?

3. In Chapter 5 we observe that the sense embeddings for senses of an ambiguous word can be socially biased. We aim to identify different types of social biases for ambiguous words, addressing the following research question:

   Can we create a benchmark dataset and metrics to evaluate social biases in pretrained sense embeddings, even if no biases are found at the word level?

4. In Chapter 6 we investigate whether the ℓ_2 norm of a static sense embedding encodes information related to the frequency of that sense in the training corpus. Specifically, we ask the following question:

   What is the relationship between ℓ_2 norm of a sense embedding and the frequency of the corresponding sense?

1.3 Contributions

The primary goals of this thesis are (a) to develop approaches to learning sense embeddings in order to disambiguate various senses of ambiguous

words and (b) to investigate the properties of sense embeddings, such as their social biased and relationship to sense frequencies. To this end, the thesis makes a number of noteworthy contributions as listed below:

1. **Context Derived Embeddings of Senses (CDES).** Contextualised word embeddings generated from NLMs, such as BERT, represent a word with a vector that considers the semantics of the target word as well as its context. However, training an NLM from scratch is expensive and time-consuming. Moreover, contextualised embeddings are not directly associated with semantic networks, such as WordNet or BabelNet. We address this limitation by first extracting sense-related information encoded in contextualised word embeddings and then injecting it into pretrained sense-agnostic static word embeddings to create static sense embeddings. CDES learns sense-specific projection matrices that can be used to predict the sense embeddings of words from their word embeddings. CDES can be seen as using contextualised language models as a proxy for extracting information relevant to a particular task, without learning it directly from text corpora. CDES is computationally relatively lightweight because it uses *pretrained* static embeddings as well as contextualised embeddings from a *pretrained* NLM and does not require training these models from scratch. In addition, CDES embeddings can be precomputed because of their independence in the context. This work has been published as a long paper at the 35^{th} Pacific Asia Conference on Language, Information and Computationis (PACLIC 35) and is further discussed in Chapter 3.

2. **Neighbour Preserving Meta-Sense Embeddings** (**NPMS**). Not all existing sense embeddings cover all senses of ambiguous words equally well due to the discrepancies in their training resources. We address this problem by incorporating multiple inde-

pendently pretrained **source** sense embeddings to learn a **meta**-sense embedding. NPMS is able to combine full-coverage sense embeddings with partial-coverage ones to improve the sense coverage in partial-coverage sense embeddings. By using NPMS, the sense-related information captured by the source (input) sense embeddings can be preserved in the (output) meta-sense embedding. This work was done in collaboration with Haochen Luo (as the final year undergraduate project at the Department of Computer Science, University of Liverpool). I have obtained consent from Mr. Luo to describe this work in my thesis, where my main contributions are as follows: (a) I was involved in the process of development of ideas, project meetings from the inauguration of this project, (b) I provided access to the pre-trained static sense embeddings used in this project as well helping Mr. Luo to meta-embed the sources using SemCor data, (c) I assisted with the evaluation of meta sense-embeddings produced in this project using WSD and WiC benchmark datasets and evaluation tools, and (d) I wrote and commented on significant portions of the research paper describing the work done in this project which has been Accepted in the Findings of the 61^{st} Annual Meeting of Association of Computational Linguistics (ACL). This work is further discussed in Chapter 4.

3. **Conducting the first ever systematic evaluation of social biases in sense embeddings.** In contrast to the numerous prior work evaluating the social biases in pretrained word embeddings, the biases in sense embeddings have been relatively understudied. To address this gap, we investigate different types of social biases in sense embeddings. Specifically, to evaluate social biases in static sense embeddings, we extended previously proposed benchmarks for evaluating social biases in static (sense-insensitive) word

embeddings by manually assigning sense ids to the words considering their social bias types expressed in those datasets. Moreover, to evaluate social biases in sense-sensitive contextualised embeddings, the SSSB dataset is created. SSSB is a novel template-based dataset containing sentences annotated for multiple senses of an ambiguous word considering its stereotypical social biases. This work has been published as a long paper at the 60^{th} Annual Meeting of the Association of Computational Linguistics (ACL) and is further discussed in Chapter 5.

4. **We discovered that the ℓ_2 norm of a static sense embedding encodes information related to the frequency of that sense in the training corpus.** Although the relationship between static word embeddings and the frequencies of words has been studied previously, such relationship has not been discovered at the sense level. To address this gap we extend the existing relationship from word embeddings to sense embeddings. Specifically, we showed that the squared ℓ_2 norm of a static sense embedding is proportional to the log frequency of the sense in the training corpus. This finding can be seen as an extension of a previously known relationship from word embeddings to sense embeddings. In addition, we find that the relationship holds for different types of static sense embeddings learned using methods such as GloVe (Pennington et al., 2014) and skip-gram with negative sampling (SGNS; Mikolov et al., 2013b) on SemCor (Miller et al., 1993). This work has been published as a short paper in the Findings of the 2022 Conference on Empirical Methods (EMNLP) and is further discussed in Chapter 6.

1.4 Publications

The majority of the materials in this thesis have been published in peer-reviewed conferences in NLP. Below is a list of publications in reverse chronological order.

- Xiaohang Tang, Yi Zhou and Danushka Bollegala: Learning Dynamic Contextualised Word Embeddings via Template-based Temporal Adaptation, Proc. of the 61^{st} Annual Meeting of Association of Computational Linguistic (ACL), 2023.

- Haochen Luo, Yi Zhou and Danushka Bollegala: *Together We make Sense*–Unsupervised Learning of Meta-Sense Embeddings, Proc. of the 61^{st} Annual Meeting of Association of Computational Linguistic (Findings of ACL), 2023. Chapter 4

- Saeth Wannasuphoprasit, Yi Zhou and Danushka Bollegala: Solving Cosine Similarity Underestimation between High Frequency Words by ℓ_2 Norm Discounting, Proc. of the 61^{st} Annual Meeting of Association of Computational Linguistic (Findings of ACL), 2023.

- Yi Zhou and Danushka Bollegala: On the Curious Case of ℓ_2 norm of Sense Embeddings, Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP), pp. 2593–2602, 2022. Chapter 6

- Yi Zhou, Masahiro Kaneko and Danushka Bollegala: *Sense Embeddings are also Biased* – Evaluating Social Biases in Static and Contextualised Sense Embeddings, Proc. of the 60^{th} Annual Meeting of Association of Computational Linguistic (ACL), Volume 1: Long Papers, pp. 1924-1935, 2022. Chapter 5

- Yi Zhou and Danushka Bollegala: Learning Sense-Specific Static Embeddings using Contextualised Word Embeddings as a Proxy, Proc. of the 35^{th} Pacific Asia Conference on Language, Information and Computation (PACLIC), pp. 588-597, 2021. Chapter 3
- Yi Zhou and Danushka Bollegala: Predicting the Quality of Translation without an Oracle, In Communications in Computer and Information Science (CCIS), Springer International Publishing, pp.3-23, 2020.
- Guanqun Cao, Yi Zhou, Danushka Bollegala and Shan Luo: Spatio-temporal Attention Model for Tactile Texture Recognition, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9896-9902, 2020.
- Yi Zhou and Danushka Bollegala: Unsupervised Evaluation of Human Translation Quality, Proc. of the 11^{th} International Conference on Knowledge Discovery and Information Retrieval (KDIR), pp. 55-64, 2019.

1.5 Thesis Outline and Summary

In this chapter, we gave an overview of the research undertaken in this thesis, a description of research aims and motivations, questions, contributions and an outline of the structure of the subsequent chapters.

Figure 1.1 organises chapters according to the topics covered in this thesis. Chapters are grouped according to sense embeddings learning methods (Chapter 3 and Chapter 4), analysis of properties of sense embeddings (Chapter 5 and Chapter 6).

Next, in Chapter 2, we provide an overview of the relevant literature on different types of word and sense embedding work, word sense disambiguation, meta embeddings and dynamic contextualised embeddings.

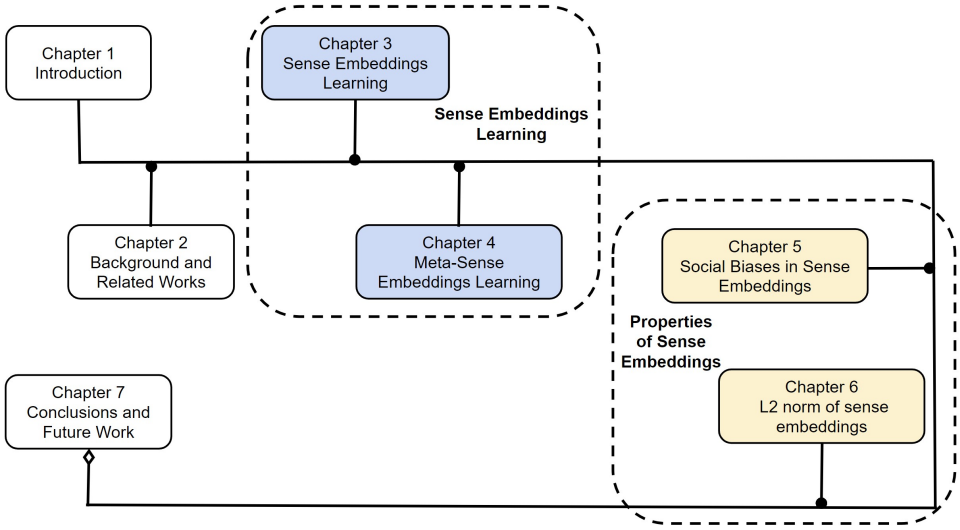


Figure 1.1: Outline of the thesis. Blue boxes indicate chapters focusing on sense embeddings learning, where yellow boxes for chapters represent on analysis properties of sense embeddings and orange is on dynamic embeddings learning.

We hope this will set the context and provide the necessary background for understanding the remainder of the thesis.

2 Background and Related Work

In this chapter, we provide an overview of the literature for static word embeddings (§ 2.1), language models (§ 2.2), sense embeddings and word sense disambiguation (§ 2.3), meta embeddings (§ 2.4) and social biases in embeddings (§ 2.5).

Given the flexibility and diversity of natural languages, representing text efficiently has always been a challenging task. The vector space model is important as it is the fundamental text representation method in NLP, which enables the application of mathematical concepts in linear algebra and statistics. In addition, vector representations are required for a wide range of machine learning algorithms and approach to handle NLP tasks. The vector representations of words, known as word embeddings, are created based on the distributional hypothesis (Harris, 1954), which states that words occurring in the same context tend to have similar meanings. With this synopsis of word embeddings we now review prior work, starting with static word embeddings.

2.1 Static Word Embeddings

Static word embeddings are a form of distributed semantic representations. With time, static word embeddings have emerged as a topic of re-

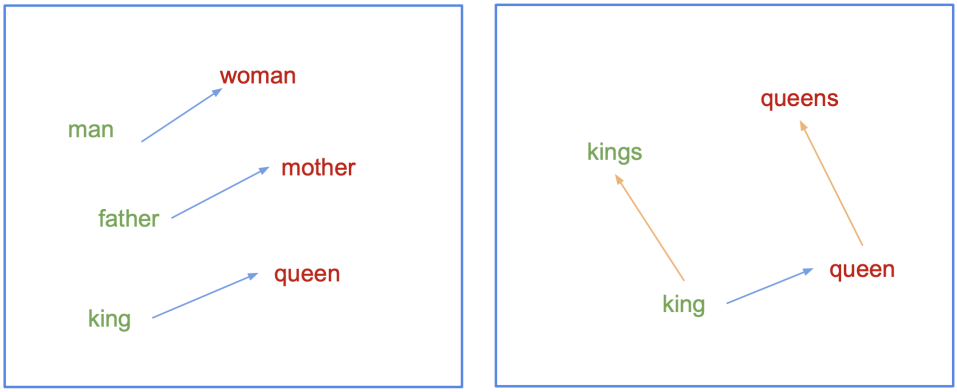


Figure 2.1: Projection of word embeddings in 2D: the left panel shows the gender relation of three word pairs, while the right panel shows the singular/plural relation. The blue arrow indicates the gender relation and the orange arrow indicates the singular/plural relations (Mikolov et al., 2013c).

search with the awareness that they can be used as standalone features. Surprisingly, they are capable in encoding not only syntactic but also semantic relations when integrated into a neural network architecture, which has been proven to be important for achieving state-of-the-art performance in many NLP tasks (Zou et al., 2013; Bordes et al., 2014; Weiss et al., 2015). Mikolov et al. (2013c) showed that all pairs of words sharing a particular relation are related by the same constant offset in the embedding space, as illustrated in Figure 2.1

Word embeddings can be commonly divided into two types according to the strategies used to produce them (Almeida and Xexéo, 2019): (1) count-based models and (2) prediction-based models. Count-based models take into account the global co-occurrence information and corpus-wide statistics such as word counts and frequencies. On the other hand, prediction-based models use local data (i.e., the context where a word occurs).

2.1.1 Count-based Models

The count-based models leverage the counts of word-context co-occurrences globally in a corpus, which are represented using word-context matrices (Turney and Pantel, 2010).

The first count-based model, namely Latent Semantic Analysis (LSA), was proposed by Deerwester et al. (1990), where Singular Value Decomposition (SVD) is applied to a term-document matrix (i.e., factorised matrix) for information retrieval (IR). Similar to producing document embeddings in IR, one can obtain word embeddings by taking the rows of the factorised matrix.

Later, Lund and Burgess (1996) proposed Hyperspace Analogue to Language (HAL), which captures the statistical dependencies between words by means of co-occurrence information in a corpus. They calculated the co-occurrence counts between each target word and its co-occurring word in all the contexts in which it appears. Then the word embedding of a target word is represented by a vector of other words that are co-occurring with it in a sliding window. However, as they do not require normalisation to be applied to the word co-occurrence counts, high frequent words, such as *the, to* and *is* etc., tend to contribute disproportionately to all the other words that co-occur with them.

To tackle this issue, Rohde et al. (2006) introduced the Correlated Occurrence Analogue to Lexical Semantic (COALS) method, which is based on HAL, but achieves better performance by using normalisation strategies to factor out lexical frequency to reduce the undue effects of high frequent neighbours of target words. Later on, Dhillon et al. (2011) contributed to count-based models by proposing Low Rank Multi-View Learning (LR-MVL) where embeddings are produced using Canonical Correlation Analysis (CCA) (Hotelling, 1953) between the past and future views of low rank approximations of the data. Lebret and Collobert

(2014) demonstrated that word embeddings can be generated by computing the Hellinger PCA of the word co-occurrence matrix. The word co-occurrence matrix can be derived by simply counting co-occurring words in a large text corpus.

Another prominent word embedding learning method is GloVe (Pennington et al., 2014), which is a log-bilinear regression model combining both global matrix factorisation and local context window methods to form word embeddings. GloVe is trained on a global word to word co-occurrence matrix rather than on the entire sparse matrix or on individual context windows in a large corpus. Therefore, GloVe is able to capture semantic relationships between words from a constructed co-occurrence matrix.

2.1.2 Prediction-based Models

Prediction-based models are produced from NLMs. In the early research, word embeddings were generated as an interesting by-product of training NLMs (Bengio et al., 2003). Specifically, word embeddings are regarded as simply the projection of raw word vectors to an embedding layer of NLMs (i.e., the first layer of NLMs). Collobert and Weston (2008) were the first to decouple word embeddings learned from downstream training objectives. Different from previous methods that only used the preceding context to train NLMs, they proposed to predict the target word by means of the complete context of a word (i.e., both the preceding and succeeding contexts of the centre word).

Subsequently, using NLMs to learn useful word embeddings attracted wide attention. Two models related to word2vec (Mikolov et al., 2013b) were proposed, namely Continuous Bag of Words (CBOW) and skip-gram with negative sampling (SGNS). The general architecture of the CBOW and SGNS is shown in Figure 2.2. These two models take the form of a simple single-layer feed-forward neural network language

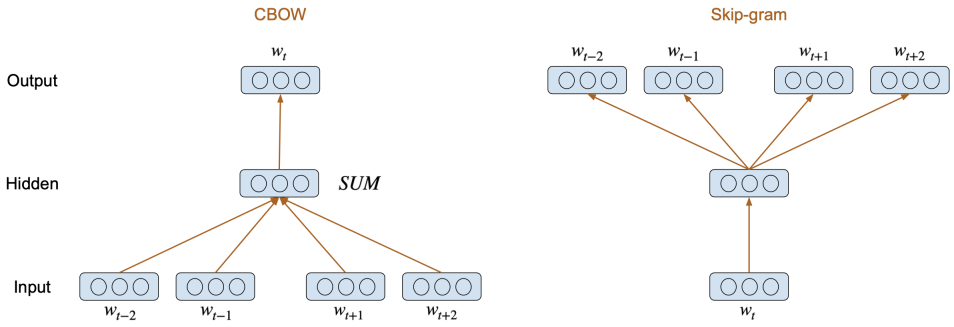


Figure 2.2: Architectures of CBOW and SGNS models (Mikolov et al., 2013a). In the CBOW model, the input layer takes the context of the centre word (i.e., target word) as a combination of one-hot representations of its surrounding words, whereas the Skip-gram model takes centre word as an input.

model based on the inner product between two word vectors. Both of them are log-linear models, whereas the main difference between them is in the training objectives. In the CBOW model, the target word is predicted using its surrounding words in the context by minimising the negative log loss function as shown in 2.1.

$$E = -\log\left(p\left(\boldsymbol{w}_t \mid \boldsymbol{W}_t\right)\right) \tag{2.1}$$

Here, $\boldsymbol{w}_t$ is the target word in the context, $W_t = w_{t-n}, \ldots, w_t, \ldots, w_{t+n}$. On the contrary, the goal in the SGNS is to predict the surrounding words using the target word. In both SGNS and CBOW models, context refers to a fixed-length window sliding over tokenised text containing the target word in the centre. Dense vector-based representations of words can then be produced with the two training objectives.

In order to improve the accuracy of the representations of less frequent words as well as speed up the training process, the models used subsampling of frequent words to reduce the amount of noise during

training (Mikolov et al., 2013b). Later on, Levy and Goldberg (2014) showed that Skip-gram can be seen as an implicit factorisation of a Point-Mutual Information (PMI) co-occurrence matrix.

As discussed in Chapter 1, word embeddings suffer from the limitation of *meaning conflation deficiency* around word types, and are insensitive to contexts owing to the fact that they reduce NLM to fixed representations. Bojanowski et al. (2017) proposed FastText, which is able to derive representations for unseen word types during training rather than being restricted to a finite set of representations. They improved the Skip-gram model by learning n-gram embeddings. The intuition is that for highly inflected languages, such as Spanish, Polish and Finnish, which heavily rely on morphology and composition in word-building, there is some information encoded in the word types, which can help to generalise to unseen words.

2.2 Contextualised Embeddings

Although static word embeddings have been proven effective in NLP tasks, they are mostly trained using shallow models and often fail to capture higher-level concepts in different contexts, for instance, ambiguity in polysemous words, syntactic structures and semantic roles. The embeddings that are derived by training NLMs are regarded as contextualised embeddings. The key difference between contextualised and static embeddings is that contextualised embeddings are sensitive to the contexts in which a word occurs. This allows the same word type to have different representations according to the contexts where it occurs. To derive various representations induced by different contexts, contextualised embeddings leverage pretrained NLMs for inferences. Moreover, the objective of NLMs to obtain contextualised embeddings is directional, which predicts the previous and/or next tokens in sentences. Given a sequence of words (i.e., a sentence) $C = w_1, w_2, \ldots w_n$, the

objective of contextualised embedding learning is to compute the contextualised word representation of w_i.

Yang et al. (2019) claimed that there are two main approaches for learning contextualised embeddings: autoregressive and autoencoding. Autoregressive language modeling seeks to estimate the probability distribution of a text corpus with an autoregressive model. In comparison, rather than performing explicit density estimation, autoencoding based models aim to reconstruct the original data from a corrupted version of the input. We will describe each of these approaches in detail in the following sections.

2.2.1 Autoregressive Models

Inspired by n-gram language models (LMs), which predict the upcoming word in a sentence given n preceding words (Jozefowicz et al., 2016), autoregerssive models extend n-gram LMs by representing previous words using context vectors and only consider relevant words in the preceding context (Torregrossa et al., 2021).

The early autoregressive models created the output distribution for words using the combination of pretrained static embeddings, Recurrent Neural Network (RNN) and softmax to represent temporal sequence with a hidden vector, and the sequence is reversed to conduct bidirectional language understanding (Akbik et al., 2018; Jozefowicz et al., 2016; McCann et al., 2017; Peters et al., 2018). Devlin et al. (2019); Yang et al. (2019) claimed that the contextualised embeddings derived using autoregressive models take into account the left and right contexts independently, whereas autoencoding models are able to perform inference with a unified context, which results in a better understanding of natural languages.

In order to handle the out-of-vocabulary (OOV) words, character convolutional neural network (CNN) (Jozefowicz et al., 2016; McCann

et al., 2017) and Byte Pair Encoding (BPE) (Sennrich et al., 2016) are leveraged to comprehend the morphology of words. Another solution is to train bidirectional models with characters rather than words. The trained static embeddings are then concatenated to perform the representation of a word.

To model both (1) complex characteristics of word use (e.g., syntax and semantics) and (2) how the meanings of words vary across contexts (i.e., ambiguity), Peters et al. (2018) proposed ELMo (Embeddings from Language Models), which is the implementation of autoregressive contextualised embeddings. ELMo achieves large improvements over a wide range of NLP tasks compared with static word embedding models. It derives contextualised embeddings for each token by concatenating the hidden states of a 2-layer biLSTM trained on a bidirectional language modelling task. The bidirectional LM is beneficial to look at a word from both left and right contexts to capture the contextual information of the word in given a sentence. In addition, ELMo embeddings are benefits from subword units, as the embeddings are entirely based on characters, which makes the network effective for the tokens that are not found in the training corpus.

FLAIR (Akbik et al., 2018) is another autoregressive contextualised word embedding model using RNN on the basis of characters. To take into account the sequence of characters, FLAIR leverages the space character to determine the boundaries of words. Both forward and backward hidden states are concatenated to unify the context into a single vector for each word. Akbik et al. (2018) showed that better performance is attained by concatenating the FLAIR embeddings with static word embeddings as static word embeddings are able to bring word-level information to the character-based FLAIR embeddings.

The more recent autoregressive models, such as GPT (Generative Pre-trained Transformer) (Radford et al., 2018, 2019; Brown et al., 2020), broadly use transformer decoder rather than long short-term

memory networks (LSTMs) or recurrent network. Using transformer has proved to be more effective when the number of parameters increases compared to the previous autoregressive models. GPT models (Radford et al., 2018, 2019; Brown et al., 2020) have taken NLP community by storm. GPT models use a transformer decoder to comprehend sequences of words. GPT-3 is the most powerful model among the GPT variants, and paving the way for future contextualised embedding models. Future NLP LMs can be developed to expand emails, generate codes and extract entities from texts according to natural language instructions with a few demonstration examples (Liu et al., 2022). GPT-3 is able to be directly applied to NLP tasks without any gradient updates or fine-tuning. It obtains competitive performance on several NLP tasks by leveraging in-context learning to concatenate the original input with task descriptions and a few examples.

2.2.2 Autoencoding Models

Autoencoding models encode bidirectional context to predict a target word, which is hidden by a [MASK] token in a sentence. Unlike autoregressive models which are unidirectional (i.e., to predict data from the previous input), autoencoding models have the capability to consider both left and right contexts by reconstructing the original context from the corrupt input (Huy et al., 2022). Autoencoding models are pretrained on a vast amount of raw texts to attain contextualised representations of the whole sentences. Such models can be fine-tuned and achieve SoTA performance in many downstream tasks, such as question answering, text generation, sentence classification and token classification (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Clark et al., 2020; He et al., 2020b).

BERT (Bidirectional Encoder Representations from Transformers) proposed by Devlin et al. (2019) is a typical autoencoding model. The

key idea of BERT is using a deep bidirectional Transformer (Vaswani et al., 2017), which allows the model to consider the left and right contexts of tokens. BERT is trained using **mask language model** pre-training objective, which is inspired by the Cloze task (Taylor, 1953). The Masked Language Model (MLM) reconstructs the input by randomly masking some of the tokens and aims to predict the original vocabulary id of the masked word based on its context. In addition to MLM, BERT uses the **next sentence prediction** task as well, which jointly learns text-pair representations.

RoBERTa (Robustly optimized BERT approach) introduced by (Liu et al., 2019) is another autoencoding model. RoBERTa uses the same architecture as BERT. However, the main difference between RoBERTa and BERT is that RoBERTa removes the **next sentence prediction** task and is trained with bigger batches over more data. In addition, RoBERTa uses dynamic masking, wherein for different epochs different parts of the sentences are masked, while BERT uses static masking.

Following the success of BERT and RoBERTa, He et al. (2020b) proposed DeBERTa (Decoding-enhanced BERT with disentangled attention), which improves BERT and RoBERTa using two techniques: **disentangled attention mechanism** and **enhanced mask decoder**. As DeBERTa represents each word with two vectors to encode its semantic information and relative positions, respectively, **Disentangled attention mechanism** computes the attention weights among words using disentangled matrices based on the two vectors. Then an **enhanced mask decoder** is used to incorporate absolute positions in the decoding layer to predict the masked tokens in model pre-training.

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) introduced by Clark et al. (2020), which leverages a more sample-efficient pre-training task called replaced token detection. Instead of masking the input as in the aforementioned models, ELECTRA corrupts the input by replacing some tokens with

plausible alternatives sampled from a small generator network rather than masking it and trains a discriminative model to predict whether each token in the corrupted input was replaced by a generator sample or not.

DistilBERT (A distilled version of BERT) proposed by Sanh et al. (2019) leverages dynamic masking, initialised the student weights with teacher weights and removes the next sentence prediction objective. DistilBERT is able to reduce the size of a BERT model by 40% by using knowledge distillation during the pre-training phase while retaining 97% of its language understanding capabilities and being 60% faster.

ALBERT (A LITE BERT) is introduced by Lan et al. (2020). ALBERT incorporates two parameter reduction techniques to lower memory consumption and speed up the training of BERT. The first one is a factorized embedding parameterization, which reduces the size of the vocabulary embeddings. The second one is cross-layer parameter sharing, which reduces the number of parameters without a performance drop and further improves performance by replacing next sentence prediction with an inter-sentence coherence loss. ALBERT produces SoTA results on natural language understanding benchmarks, such as GLUE (Wang et al., 2018), SQuAD (Rajpurkar et al., 2016), and RACE (Lai et al., 2017).

2.3 Word Sense Disambiguation and Sense Embeddings

Word Sense Disambiguation (WSD) is a historical task in NLP and AI, which dates back to Weaver (1945), who discovered the problem of polysemous words in the context of Machine Translation. Nowadays, word polysemy remains one of the most challenging linguistic phenomena in NLP. Researchers have long sought ways to handle such phenomenon

with the task of WSD, which is at the forefront of the automatic resolution of polysemy (Bevilacqua et al., 2021). In WSD task, ambiguity is tackled by mapping a target word to one (or potentially more) of its possible senses according to the surrounding context.

2.3.1 Resources for WSD

Performing WSD requires two different kinds of data:

- Sense inventories: lexical resources, such as a dictionary or thesaurus, that list different meanings (senses) of each word.
- Annotated corpora: in which a subset of words are tagged with one or more possible meanings (senses) according to the given inventory.

In the following subsections, we review the most popular sense inventories (§ 2.3.1.1) and annotated corpora (§ 2.3.1.2) used for training and testing WSD systems.

2.3.1.1 Sense Inventories

Sense inventories list the set of possible senses for each given word. In this section, we describe the most commonly used ones.

WordNet (Fellbaum and Miller, 1998) is one of the most commonly used sense inventories. It is a manually-curated lexicographic database of English. WordNet is structured as a graph where nodes are synsets, i.e., groups of synonymous words that correspond to the same sense. Each synonym in a synset represents a sense of a word. Synsets and senses are linked to each other via edges, which represent lexical or semantic relations, such as hypernymy (is-a), and meronymy (part-of), among others. Specifically, hypernymy connects more general synsets, for example {*furniture, piece_of_furniture*} to increasingly specific ones,

such as {*bed*} and {*bunkbed*}. On the other hand, meronymy is the part-whole relation, which holds between synsets, for instance chair and back, backrest, seat and leg. For each synset, WordNet provides other forms of lexical knowledge as well, such as definitions (glosses) and usage examples. The most recent version of WordNet is version 3.1 released in 2012, which covers 155,327 words and 117,979 synsets.

BabelNet proposed by Navigli and Ponzetto (2012) is a multilingual inventory, which covers both lexicographic and encyclopedic terms obtained by semi-automatically mapping various resources, such as WordNet, multilingual versions of WordNet, Wikipedia, Wikidata and Wiktionary, to name a few. BabelNet is organised as a semantic network where nodes represent multilingual synsets, and edges are semantic relations between them. The latest version of BabelNet is version 5.1, which covers 500 languages and includes more than 20M synsets.

2.3.1.2 Sense-Annotated Data

Several sense-annotated datasets have been created so far. Here, we limit our discussion to the standard datasets that are used in WSD.

Training Data: SemCor (Miller et al., 1993) is the most commonly used training data for WSD. Words in SemCor are annotated with PoS tags, lemmas, and word senses from the WordNet inventory. Overall, SemCor contains more than 200,000 sense annotations, thus acting as the largest sense-tagged corpus for training sense classifiers with supervised disambiguation settings. However, SemCor only covers 15% of the synsets in WordNet. Being a subset of the English Brown Corpus from the 1960s, it does not contain many meanings that are now widespread. For instance, *computer mouse* (Bevilacqua et al., 2021). In order to extend the annotation coverage, Taghipour and Ng (2015) introduced OMSTI (One Million Sense-Tagged Instances) is a large corpus automatically annotated with senses using WordNet 3.0. OMSTI was created by

using an alignment-based WSD approach (Chan et al., 2005) on a large English-Chinese parallel corpus (Eisele and Chen, 2010) More recently, many works (Vial et al., 2019; Bevilacqua and Navigli, 2020) have begun leveraging the English Princeton WordNet Gloss Corpus (WNG)[1] as additional data. WNG includes WordNet sense definitions and examples that have been manually and semi-automatically annotated to cover more than 60,000 WordNet senses. Even though English training data is widely accessible, this is not the case for other languages. Manually annotated data are notoriously difficult to obtain on a large scale for many languages. It is costly and difficult to find native speakers in all languages. As a result, several subsequent works proposed automated methods for creating high-quality sense-annotated data in both English (Petrolito and Bond, 2014; Loureiro and Camacho-Collados, 2020) and other languages by utilising data from Wikipedia (Scarlini et al., 2019), the Personalized PageRank algorithm (Pasini and Navigli, 2020), label propagation over similar texts (Barba et al., 2021b), or automatic translations (Pasini et al., 2021).

Testing Data: The manually annotated datasets from the Senseval and SemEval evaluation campaigns are often used for evaluation in WSD. The evaluation framework proposed by Raganato et al. (2017), which is used for English WSD, consists of five all-words gold-standard datasets from the Senseval and SemEval competitions:

- Senseval-2 (**SE2**; Edmonds and Cotton, 2001) was originally annotated using WordNet 1.7. SE2 contains 2,282 sense annotations, including nouns, verbs, adverbs and adjectives.

- Senseval-3 (**SE3**; Snyder and Palmer, 2004) consists of three documents from three different domains (editorial, news story and

[1] https://wordnetcode.princeton.edu/glosstag.shtml

fiction), and contains 1,850 sense annotations in total. SE3 was annotated with WordNet version 1.7.1.

- SemEval-07 (**SE07**; Pradhan et al., 2007) is the smallest one over the five datasets, which consists of 455 sense annotations for nouns and verbs only. SE07 was originally annotated with WordNet 2.1 sense inventory.

- SemEval-13 (**SE13**; Navigli et al., 2013) comprising 13 documents from different domains. SE13 was originally annotated using WordNet 3.0. The number of sense annotations is 1,644, and only nouns are considered.

- SemEval-15 (**SE15**; Moro and Navigli, 2015) is the most recent WSD dataset available to date, annotated with WordNet 3.0. SE15 includes 1,022 sense annotations in 4 documents from three heterogeneous domains: biomedical, mathematics/computing and social issues.

The evaluation of English WSD with the WordNet sense inventory was standardised by this framework, making it simpler to compare systems in a general domain and advancing the development of ever-better models.

More recently, a comprehensive benchmark has been proposed to standardise the evaluation in multilingual setting (**XL-WSD**; Pasini et al., 2021). XL-WSD extends the English evaluation framework of Raganato et al. (2017) and creates test data for 18 languages resulting in more than 99K gold annotations. This benchmark enables a large-scale monolingual and multilingual evaluation for WSD models, including the cross-lingual zero-shot setting, i.e., training in English and testing in other languages. The training and testing data are annotated with BabelNet 4.0 senses.

2.3.2 Approaches to WSD

Through decades of research, different solutions to solve WSD task has been proposed, which can be generally divided into two classes: **supervised** approaches (§ 2.3.2.1) and **knowledge-based** ones (§ 2.3.2.2).

2.3.2.1 Supervised WSD

Supervised approaches leverage machine learning methods to induce a classifier from manually sense-annotated data sets, and aim to train a parameterised function f to map a word w in a context c to a sense s using sense annotated corpora. The classifier, also known as a word expert, typically focuses on a single word and conducts a classification task to assign the proper sense to each instance of that word (Navigli, 2009). The training set used to develop the classifier comprises a set of instances where a given target word is manually assigned a sense based on the sense inventory. The most meaningful classification of the approaches is concerned with what kind of extra information the model is able to use rather than the architecture, therefore supervised approaches can be further grouped into four categories (Bevilacqua et al., 2021): (1) purely data-driven models (Hadiwinoto et al., 2019; Bevilacqua and Navigli, 2019; Vial et al., 2019), (2) supervised models exploiting glosses (Huang et al., 2019; Loureiro and Jorge, 2019a; Kumar et al., 2019; Scarlini et al., 2020a,b; Wang and Wang, 2020; Yap et al., 2020; Bevilacqua and Navigli, 2020; Blevins and Zettlemoyer, 2020), (3) supervised models exploiting relations in a knowledge graph (Kumar et al., 2019; Loureiro and Jorge, 2019a; Vial et al., 2019; Scozzafava et al., 2020; Bevilacqua and Navigli, 2020), and (4) supervised approaches using other sources of knowledge (Calabrese et al., 2021; Scarlini et al., 2020a,b; Wang and Wang, 2020). Despite the high demand for a huge labeled corpus, the majority of supervised approaches are more effective than those in the knowledge-based category and achieve the cur-

rent SoTA performance (Blevins and Zettlemoyer, 2020; Bevilacqua and Navigli, 2020; Yap et al., 2020; Barba et al., 2021a) in WSD.

2.3.2.2 Knowledge-based WSD

Knowledge-based approaches make use of graph algorithms on a semantic network such as WordNet or BabelNet, in which synsets are considered as nodes and the relationships between them as edges. The successful knowledge-based approaches use graph algorithms such as random walks (Agirre et al., 2014), clique approximation (Moro et al., 2014), or game theory (Tripodi and Navigli, 2019). The performance of such systems mostly depends on the quantity and quality of the information embedded inside their underlying knowledge bases (Pilehvar and Navigli, 2014; Maru et al., 2019). There are two research streams in knowledge-based approaches. One is to take into account overlap or similarity between the context containing a word under disambiguation and the relevant information (e.g., the definition of a potential sense) from a Knowledge Base (KB). The most similar sense is selected as the predicted one. The other one is to construct a graph using the provided context and all connections retrieved from some KBs. The sense of a particular word is then predicted using the built graph by applying different graph-based algorithms, such as PageRank (Brin and Page, 1998) and Latent Dirichlet Allocation (**LDA**; Blei et al., 2003). Even though these approaches do not perform as good as supervised ones, they have wider sense coverage due to the use of large-scale knowledge resources (Navigli, 2009).

2.3.3 Sense Embeddings Learning Approaches

Many efforts have been made to learn multi-prototype embeddings for different senses of words. The main idea for doing so is to augment the standard word embeddings by disambiguating word senses according

to the contexts in that a word appears. Reisinger and Mooney (2010) proposed multi-prototype embeddings to represent word senses, which was extended by Huang et al. (2012) combining both local and global contexts. Both methods use clustering to group contexts of a word related to the same sense. Even though the number of senses depends on the word, these methods assign a fixed number of senses to all words. To overcome this limitation, Neelakantan et al. (2014) proposed a non-parametric model, which estimates the number of senses dynamically per word.

Although clustering-based methods assign multi-prototype embeddings for a word, they still have a drawback in that the trained embeddings are not associated with any sense inventories (Camacho-Collados and Pilehvar, 2018b). In contrast, knowledge-based approaches learn sense embeddings by extracting sense-specific information from external sense inventories, such as WordNet and BabelNet. Chen et al. (2014) extended word2vec to learn sense-specific embeddings associated with the WordNet (Fellbaum and Miller, 1998) *synsets*. Rothe and Schütze (2015) used the semantic relations in WordNet to embed words and their senses into a common vector space. Iacobacci et al. (2015) used the sense definitions from BabelNet and perform WSD to obtain sense-specific contexts.

Recently, contextualised embeddings generated by NLMs have been used to create sense embeddings. Loureiro and Jorge (2019a) proposed LMMS (Language Modelling Makes Sense) for constructing sense embeddings by taking the average over the contextualised embeddings of the sense annotated tokens from SemCor. Scarlini et al. (2020a) introduced SenseEmBERT (Sense Embedded BERT) using the lexical-semantic information in BabelNet to produce sense embeddings without relying on sense-annotated data. Scarlini et al. (2020b) also proposed ARES (context-AwaRe EmbeddinS), a knowledge-based approach for constructing BERT-based embeddings of senses by means of the lexical-

semantic information in BabelNet and Wikipedia.

In Chapter 3, we propose a method for injecting sense-related information into static word embeddings using pretrained contextualised word embeddings. In Chapter 6 we study the relationship between the ℓ_2 norm of a sense embedding and the frequency of the corresponding sense and find that the ℓ_2 norm of a sense embedding is a surprisingly effective feature for WSD.

2.4 Meta Embeddings

To date, numerous methods for learning word embeddings that take into account various features of semantics such as context, sense, and multilinguality have been developed. Meta-embedding (ME) learning is an approach for learning more accurate word embeddings using only existing (source) word embeddings as input. It was first proposed for combining multiple pretrained static word embeddings (Yin and Schütze, 2016). Nowadays, ME learning has gained attention in the NLP community due to its capacity to include semantics from numerous source embeddings in a compact manner with better performance. The input and output word embeddings to the ME algorithm are referred to as the source and meta-embeddings, respectively. Given a set of N source word embeddings $s_1, s_2, \ldots, s_N$ respectively covering vocabularies (i.e. sets of words) $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n$. The embedding of a word w in s_j is denoted by $\boldsymbol{s}_j(w) \in \mathbb{R}^{d_j}$, where d_j is the dimensionality of s_j. s_j can be represented by an embedding matrix $\mathbf{E}_j \in \mathbb{R}^{d_j \times |\mathcal{V}_j|}$. Then, the problem of ME is to find an optimal to combine $\mathbf{E}_1, \ldots, \mathbf{E}_n$ in order to maximise some goodness measure established for the accuracy of the semantic representation for the words. According to Bollegala and O'Neill (2022), ME learning methods can be divided into 4 categories:

Unsupervised Meta-Embedding Learning: Unsupervised ME learning methods (Bao and Bollegala, 2018; Bollegala et al., 2018; Jawan-

puria et al., 2020) do not require manually annotated labeled data in the learning process. In this setting, all data is limited to the pretrained source embeddings. Unsupervised ME learning methods can be further divided into 4 groups: concatenation, averaging, linear projections and autoencoding.

Supervised Meta-Embedding Learning: In contrast to the unsupervised ME learning approaches (O'Neill and Bollegala, 2018; Wu et al., 2020; Kiela et al., 2018; Xie et al., 2019; He et al., 2020a), supervised MEs use end-to-end learning to fine-tune the MEs specifically for downstream tasks.

Sentence-level Meta-Embedding Learning: Sentence-level ME learning methods (Poerner et al., 2020) combine sentence embeddings from pretrained encoders rather than source word embeddings.

Multi-lingual Meta-Embedding Learning: MEs have also been extended to the cross-lingual and multi-lingual settings (Winata et al., 2019b,a). These methods are done by projecting the embeddings into a common vector space. García et al. (2020) showed that the quality of learned embeddings of low-resource languages can be improved by using embeddings of resource-rich languages.

In Chapter 4, we extend prior work on word-level meta-embedding learning to learn sense-level meta-embeddings. To the best of our knowledge, this is the first-ever sense-level meta-embedding learning work that has been proposed.

2.5 Social Biases in Embeddings

Text caries characteristics of the social world since it is a medium for conveying and expressing human interactions. Text has been utilised throughout human history to not only organise and understand sociopolitical events but also to influence how these events are perceived and interpreted (Joseph, 2006). Because of the nature of text, social bi-

ases are imprinted in word embeddings. Recent research has shown that social biases such as stereotypes and prejudice present in data are amplified and contained in word embeddings (Bolukbasi et al., 2016; Garg et al., 2018). Existing methods (Caliskan et al., 2017a; Du et al., 2019) for detecting biases in word embeddings are based on qualitative concept association techniques. They investigate the qualities of groups and their relationships to other concepts, presuming that these concepts would have been assigned to these groups equally or not at all. For instance, an occupation should not be associated with one gender more than the other, nor should one gender be considered more preferable than the other.

Until now, researchers have explored different aspects of bias in word embeddings. Garg et al. (2018) demonstrated how social biases evolve over time and become encoded in word embeddings. Dev and Phillips (2019) showed that names in word embeddings serve as a proxy for bias towards social groups. Zhao et al. (2018a) introduced a method for training word embeddings that are without sexist bias. Brunet et al. (2019) proposed a method for tracing the origin of bias in embeddings back to the original text. Caliskan et al. (2017a) developed a general framework for tracing bias in word embeddings.

However, social biases in sense embeddings have not been studied in prior work, which we focus on in Chapter 5.

3 Sense Embeddings Learning

3.1 Introduction

Representing the meanings of words using low-dimensional vector embeddings has become a standard technique in NLP. *Static word embeddings* (Mikolov et al., 2013b; Pennington et al., 2014) represent words by assigning a single vector for all occurrences of a word irrespective of its *senses*. However, representing ambiguous words using a single embedding is problematic as discussed in Chapter 1.

To address this problem, *sense-specific static embedding* learning methods (Reisinger and Mooney, 2010; Neelakantan et al., 2014; Huang et al., 2012) assign multiple embeddings to a single polysemous word corresponding to its multiple senses. For example, the ambiguous word *bass* will have different embeddings corresponding to its meanings of a *musical instrument* and a type of *fish*. However, these embeddings are context-insensitive and we must resort to different heuristics such as selecting the sense embedding of the ambiguous word that is most similar to the context, in order to determine which embedding should be selected to represent the word.

On the other hand, *contextualised word embeddings* generated from NLMs (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019) repre-

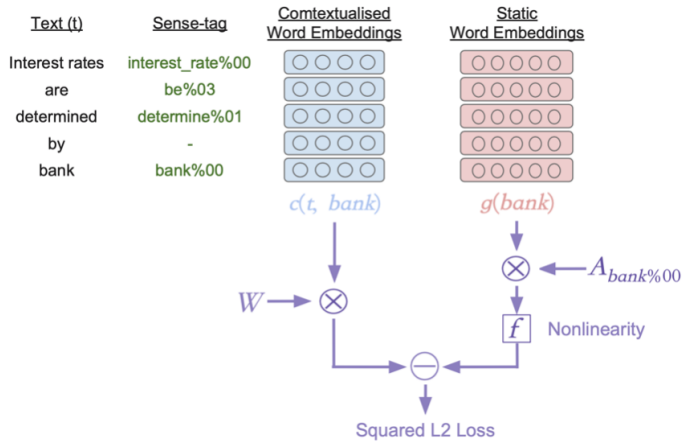


Figure 3.1: Outline of CDES. Given a sense-tagged sentence t, we compute a sense embedding for the ambiguous word *bank* by multiplying its static word embedding, $\boldsymbol{g}(bank)$, by a sense-specific projection matrix, $\mathbf{A}_{bank\%00}$, corresponding to the correct sense of the word. Projection matrices are learned by minimising the squared ℓ_2 loss between the linearly transformed (via a matrix $\mathbf{W}$) contextualised embedding, $\boldsymbol{c}(t, bank)$, and of the (nonlinearly transformed via function f) sense embedding of *bank*.

sent a word in a given context by an embedding that considers both the meaning of the word itself as well as its context. Different types of information such as word sense, dependency, and numeracy have shown to be encoded in contextualised word embeddings, providing rich, context-sensitive input representations for numerous downstream NLP applications. More recently, Loureiro and Jorge (2019a) and Scarlini et al. (2020a) showed that contextualised embeddings such as BERT and ELMo can be used to create sense embeddings by means of external semantic networks, such as WordNet and BabelNet. Moreover, Levine et al. (2020) showed that BERT can be fine-tuned using WordNet's supersenses to learn contextualised sense embeddings.

Inspired by these prior successes, we ask and affirmatively answer the question – *can we extract sense-related information from contextu-*

alised word embeddings to create sense-specific versions of (pretrained) sense-agnostic static embeddings? To this end, we propose Context Derived Embeddings of Senses (CDES), a method to extract sense-related information encoded in contextualised word embeddings and inject it into pretrained sense-agnostic static word embeddings to create sense-specific static embeddings. Given a contextualised embedding, a static word embedding and a sense-annotated corpus, CDES learns sense-specific projection matrices that can be used to predict the sense embeddings of words from their word embeddings. Following the distributional hypothesis (Harris, 1954), we require that the predicted sense embedding of a word must align (possibly nonlinearly) with the meaning of the context of the word, represented using a contextualised embedding as outlined in Figure 3.1.

At a more conceptual level, CDES can be seen as using contextualised language models as a proxy for extracting information relevant to a particular task, without learning it directly from text corpora. In particular, prior work probing language models has shown that rich information about languages is compactly and accurately encoded within contextualised representations produced by NLMs (Klafka and Ettinger, 2020). Moreover, CDES can also be seen as an instance of *model distillation* (Furlanello et al., 2018), where a complex teacher model (i.e. a contextualsied word embedding) is used to train a simpler student model (i.e. a sense-sensitive static embedding).

There are several advantages in CDES for learning sense-specific static embeddings. CDES is computationally relatively lightweight because it uses *pretrained* static embeddings as well as contextualised embeddings from a *pretrained* NLM and does not require training these resources from scratch. CDES static sense embeddings can be precomputed because of their independence on the context. Therefore, CDES embeddings are attractive to NLP applications that must run on limited hardware resources. Because subtokenisation methods, such as

BPE, must be used to limit the vocabulary sizes, one must post-process subtoken embeddings (e.g. by mean pooling) to create word embeddings with contextualised embeddings, whereas static embeddings can directly learn word embeddings. To increase the coverage of sense embeddings, in addition to the sense related information extracted from contextualised embeddings, CDES incorporates contextual information from external corpora and knowledge bases.

We evaluate CDES on WSD (Navigli, 2009) (§3.3.2) and WiC (Pilehvar and Camacho-Collados, 2019) (§3.3.3) tasks. In both tasks, CDES learns accurate sense embeddings and outperforms many existing static sense embeddings. In particular, on the WSD framework (Raganato et al., 2017), CDES reports the best performance in 4 out of 6 benchmarks, and on WiC reports competitive results to the current state-of-the-art without any fine-tuning of on task data.

3.2 Context-Derived Embedding of Senses

Given (a) pretrained static word embeddings, (b) contextualised word embeddings from a pretrained NLM, and (c) a sense-annotated corpus, CDES learns a sense-specific version of (a), representing each sense of a word by a different vector. To describe CDES in detail, let us denote the sense-agnostic static embedding of a word $u \in \mathcal{V}$ in a vocabulary $\mathcal{V}$, by $\boldsymbol{g}(u) \in \mathbb{R}^p$. Moreover, let us denote the contextualised embedding model c, from which we can obtain a context-sensitive representation $\boldsymbol{c}(u,t) \in \mathbb{R}^q$ corresponding to u in some context $t \in \mathcal{C}(u)$. Here, $\mathcal{C}(u)$ is the set of contexts in which u occurs. An ambiguous word u is likely to take different senses in different contexts t, and our goal is to learn a sense-specific embedding of u that captures the different senses of u.

Let us denote by $\mathcal{S}$ the set of word senses taken by all words in $\mathcal{V}$.

An ambiguous word u will belong to a subset $\mathcal{S}(u)$ of senses in $\mathcal{S}$. Let us denote the sense-specific embedding of u corresponding to the i-th sense $s_i \in \mathcal{S}(u)$ by $\boldsymbol{s}_i(u) \in \mathbb{R}^p$. We model the process of creating sense-specific embeddings from static embeddings as a projection learning task, where we multiply the static embedding, $\boldsymbol{g}(u)$, by a sense-specific projection matrix, $\mathbf{A}_i$, to produce $\boldsymbol{s}_i(u)$ as in (3.1).

$$\boldsymbol{s}_i(u) = \mathbf{A}_i \boldsymbol{g}(u) \tag{3.1}$$

Here, (3.1) decouples a sense embedding into a sense-agnostic static lexical semantic component given by $\boldsymbol{g}(u)$ and a word-independent sense-specific component $\mathbf{A}_i$, enabling efficient sense-specific embedding learning using pretrained embeddings. The projection matrices can be seen as linear operators that produce different sense-specific embeddings from the same static word (lemma) embedding, corresponding to the different senses of the lemma.

On the other hand, $\boldsymbol{c}(u,t)$ encodes both sense related information for u as well as information not related to u such as the grammatical gender or number in the context t. Therefore, we apply a linear filter parameterised by a matrix $\mathbf{W} \in \mathbb{R}^{q \times p}$, to extract sense related information from $\boldsymbol{c}(u,t)$.

Given a sense tagged corpus, we jointly learn $\mathbf{W}$ and $\mathbf{A}_i$s by minimising the objective given by (3.2).

$$L(\mathbf{W}, \{\mathbf{A}_i\}_{i=1}^{|\mathcal{S}|}) = \sum_{\substack{u \in \mathcal{V} \\ t \in \mathcal{C}(u) \\ s_i \in \mathcal{S}(u)}} ||\mathbf{W}\boldsymbol{c}(u,t) - f(\mathbf{A}_i \boldsymbol{g}(u))||_2^2 \tag{3.2}$$

Here, f is an elementwise nonlinear function that enables us to consider nonlinear associations between contextualised and static word embeddings. In our experiments, we consider linear, ReLU and GELU

activations as f. After training, we can compute the sense embeddings $\boldsymbol{s}_i(u)$ using (3.1) with the pretrained static word embeddings $\boldsymbol{g}(u)$.

Eq. (3.2) can be seen as aligning the contextualised and static word embeddings under a nonlinear transformation. The only learnable parameters in our proposed method are $\mathbf{W}$ and sense-specific projections $\mathbf{A}_1, \ldots, \mathbf{A}_{|\mathcal{S}|}$. In particular, we *do not* require re-training or fine-tuning static or contextualised embeddings and can be seen as a post-processing method applied to pretrained embeddings, similar to retrofitting (Shi et al., 2019). We limit the sense-specific projection matrices to diagonal matrices in our experiments because in our preliminary investigations, we did not find any significant advantage in using full matrices compared to the extra storage. Moreover, a diagonal matrix can be compactly represented by storing only its diagonal elements as a vector, which reduces the number of parameters to learn (thus less likely to overfit) and speeds up matrix-vector multiplications.

3.2.1 Context Aggregation

An important limitation of the above-mentioned setting is that it requires sense-annotated corpora. Manually annotating word senses in large text corpora is expensive and time consuming. Moreover, such resources might not be available for low resource languages. Even if such sense-annotated corpora are available for a particular language, they might not cover all different senses of all of the words in that language, resulting in an inadequate sense coverage. For example, SemCor (Miller et al., 1993), one of the largest manually-annotated corpora for English word senses including more than 220K words tagged with 25K distinct WordNet meanings, covers only 15% of all synsets in the WordNet. To address this sense-coverage problem, we follow prior proposals (Scarlini et al., 2020b) to extract additional contexts for a word from (a) **the dictionary definitions of synsets**, and (b) **an external corpus**.

Gloss-based Sense Embeddings: To create sense embeddings from dictionary definitions, we use the glosses of synsets in WordNet. Given a word u, we create a gloss-based sense embedding, $\boldsymbol{\psi}(u)_i \in \mathbb{R}^q$, represented by the sentence embedding, $\boldsymbol{c}(t_i)$, computed from the gloss t_i corresponding to the synset s_i of u. Here, $\boldsymbol{c}(t_i)$ is computed by averaging the contextualised embeddings for the tokens in the gloss t_i as given in (3.3).

$$\boldsymbol{c}(t_i) = \underset{w \in t_i}{\text{avg}}\, \boldsymbol{c}(w, t_i) \tag{3.3}$$

Here, avg denotes mean pooling over the tokens w in t_i.

Following Loureiro and Jorge (2019a) and Scarlini et al. (2020b), in our experiments, we use BERT as the contextualised embedding model and use the sum of the final four layers as token embeddings.

Corpus-based Sense Embeddings: To extract contexts from an external corpus for given a word u, we retrieve all sentences as contexts $t \in \mathcal{C}(u)$ from the corpus where u occurs. We then cluster the extracted sentences (represented by the sentence embeddings computed using (3.3)) using the k-means algorithm. We assume each cluster to contain similar sentences and that u will be used in the same sense in all sentences in a cluster. We use UKB[1] (Agirre et al., 2014), a knowledge-based approach to WSD that uses the Personalised PageRank algorithm (Haveliwala et al., 2002), to disambiguate the clusters.

To increase the coverage of senses represented by the clusters, we consider collocations of u available in SyntagNet (Maru et al., 2019)[2] following Scarlini et al. (2020b). Specifically, for each word u, we find words v that form a collocation with u in SyntagNet and extract sentences t that contain both u and v within a co-occurrence window. The

[1] http://ixa2.si.ehu.eus/ukb/
[2] http://syntagnet.org/

synset id s_i assigned to the (u, v) pair in SyntagNet is used as the sense id for all extracted sentences for u. Finally, we compute a corpus-based sense embedding $\boldsymbol{\phi}_i(u) \in \mathbb{R}^q$ as the cluster centroid, where sentence embeddings are computed using (3.3).

3.2.2 Sense Embedding and Disambiguation

The final CDES static sense embedding, $\mathbf{cdes}_i(u) \in \mathbb{R}^{p+2q}$ of the i-th sense of u is computed as the concatenation of $\boldsymbol{s}_i(u)$ (given by (3.1)), gloss-based sense embedding $\boldsymbol{\psi}_i(u)$ and corpus-based sense embedding $\boldsymbol{\phi}_i(u)$ as given by (3.4), where $\oplus$ denotes vector concatenation.

$$\mathbf{cdes}_i(u) = \boldsymbol{s}_i(u) \oplus \boldsymbol{\psi}_i(u) \oplus \boldsymbol{\phi}_i(u) \tag{3.4}$$

In order to disambiguate a word u in a given context t', we first compute a contextualised embedding $\boldsymbol{\zeta}(u, t') \in \mathbb{R}^{p+2q}$ by concatenating three vectors as give by (3.5)

$$\boldsymbol{\zeta}(u, t') = \boldsymbol{g}(u) \oplus \boldsymbol{c}(u, t') \oplus \boldsymbol{c}(u, t') \tag{3.5}$$

We then compute the cosine similarity between $\boldsymbol{\zeta}(u, t')$ and $\mathbf{cdes}_i(u)$ for each sense s_i of u. We limit the candidate senses based on the lemma and part-of-speech of u in t', and select the most similar (1-NN) sense of u as its disambiguated sense in context t'.

3.3 Experiments

In this section, we evaluate the performance of pre-trained CDES on the downstream WSD and WiC tasks.

3.3.1 Experimental Setup

In our experiments, we use the pretained GloVe[3] embeddings (Common Crawl with 840B tokens and 2.2M vocabulary) as the static word embeddings $\boldsymbol{g}(u)$ with $p = 300$. We use pretrained BERT (`large-bert-cased`[4]) as the contextualised embedding model, $\boldsymbol{c}(u, t)$ with $q = 1024$. Following prior work (Luo et al., 2018a,b; Loureiro and Jorge, 2019a; Scarlini et al., 2020b), we use sense annotations from SemCor 3.0 (Miller et al., 1993) as the sense-tagged corpus, which is the largest corpus annotated with WordNet sense ids. As the external corpus for extracting contexts as described in Section 3.2.1, we use the English Wikipedia. The number of clusters in k-means is set to the number of distinct senses for the lexeme according to WordNet. The number of words given to UKB is set to 5 and the number of sentences extracted from Wikipedia per lemma is set to 150 following Scarlini et al. (2020b). The co-occurrence window size for considering collocations extracted from SyntagNet is set to 3 according to Maru et al. (2019). We evaluate the learned sense embeddings in two downstream tasks: WSD (Section 3.3.2) and WiC (Section 3.3.3). The statistics of SemCor, all-words English WSD and WiC datasets are shown in Table 3.1.

To project contextualised and static word embeddings to a common space, we set $\mathbf{W} \in \mathbb{R}^{300\times1024}$. To reduce the memory footprint, the number of trainable parameters and thereby overfitting, we constrain the sense-specific matrices $\mathbf{A}_i \in \mathbb{R}^{300\times300}$ to be diagonal. We initialise all elements of $\mathbf{W}$ and $\mathbf{A}_i$s uniformly at random in $[0, 1]$. We use Adam as the SGD optimiser and set the minibatch size to 64 with an initial learning rate of 1E-4. All hyperparameter values were tuned using a randomly selected subset of training data set aside as a validation dataset. The t-SNE visualisations in the thesis are produced

[3] `nlp.stanford.edu/projects/glove/`
[4] `https://bit.ly/33Nsmou`

Dataset	Total	Nouns	Vebs	Adj	Adv
SemCor	226,036	87,002	88,334	31,753	18,947
WSD					
SE2	2,282	1,066	517	445	254
SE3	1,850	900	588	350	12
SE07	455	159	296	-	-
SE13	1,644	1,644	-	-	-
SE15	1,022	531	251	160	80
ALL	7,253	4,300	1,652	955	346
WiC	Instances	Nouns	Vebs	Unique Words	
Training	5,428	2,660	2,768	1,256	
Dev	638	396	242	599	
Test	1,400	826	574	1,184	

Table 3.1: The statistics of the training and evaluation datasets. SemCor is used for training. SemEval (SE07, SE13, SE15) and Senseval (SE2, SE3) datasets are used for the WSD task, whereas the WiC dataset is used for the sense discrimination task.

with `sklearn.manifold.TSNE` using n_components=2, init=*pca*, perplexity=3, n_iter=1500 and metric=*cosine*.

All experiments were conducted on a machine with a single Titan V GPU (12 GB RAM), Intel Xeon 2.60 GHz CPU (16 cores) and 64 GB of RAM. Overall, training time is less than 3 days on this machine,

3.3.2 Word Sense Disambiguation (WSD)

To evaluate the proposed sense embeddings, we conduct a WSD task using the evaluation framework proposed by Raganato et al. (2017), which is described in § 2.3.1.2. We used the framework's official scoring scripts to avoid any discrepancies in the scoring methodology. As described in § 3.2.2, the sense of a word in a context is predicted by the 1-NN method.

Table 3.2 shows the WSD results. Most Frequent Sense (MFS) baseline selects the most frequent sense of a word in the training corpus and has proven to be a strong baseline (McCarthy et al., 2007), this may be because although ambiguous words contain multiple senses, not all these senses are used equally. Especially in a given document/corpus with a specific domain, only a subset of the senses are used. Scarlini et al. (2020b) use Peters et al. (2018)'s method with BERT on SemCor+OMSTI (Taghipour and Ng, 2015) to propose SemCor+OMSTI$_{BERT}$ baseline. ELMo k-NN uses ELMo embeddings to predict the sense of a word following the nearest neighbour strategy. Specifically, they first obtain ELMo embeddings for all words in SemCor sentences, and average the embeddings for each sense. At test time, they run ELMo on the given test sentence containing the ambiguous word and select the sense with the highest cosine similarity. Loureiro and Jorge (2019a) repeated this method using BERT embeddings to propose the BERT k-NN baseline. EWISE$_{ConvE}$ (Kumar et al., 2019) learns a sentence encoder for sense definition by using WordNet relations as well as ConvE (Dettmers et al., 2018). Scarlini et al. (2020b) report the performance of using BERT base-multilingual-cased (mBERT) instead of BERT large with MFS fallback. Hadiwinoto et al. (2019) integrate pretrained BERT model with Gated Linear Unit (GLU) and Layer Weighting (LW).

GlossBERT (Huang et al., 2019) fine-tunes the pretrained BERT model by jointly encoding contexts and glosses. LMMS (Loureiro and Jorge, 2019a) learns sense embeddings using BERT to generate contextualised embeddings from semantic networks and sense definitions. To perform WSD, they use the 1-NN method and compare sense embeddings against contextualised embeddings generated by BERT. Scarlini et al. (2020b) augment UKB with SyntagNet's relations (Scozzafava et al., 2020) and obtain UKB$_{+Syn}$. SensEmBERT is a knowledge-based approach, which produces sense embeddings by means of BabelNet and Wikipedia. Although SensEmBERT is effective in modelling nominal

Models	SE2	SE3	SE07	SE13	SE15	ALL
MFS	65.6	66.0	54.5	63.8	67.1	65.6
SemCor+OMSTI$_{BERT}$	74.0	70.6	63.1	72.4	75.0	72.2
ELMo k-NN	71.5	67.5	57.1	65.3	69.9	67.9
BERT k-NN	76.3	73.2	66.2	71.7	74.1	73.5
EWISE$_{ConvE}$	73.8	71.1	67.3	69.4	74.5	71.8
mBERT k-NN + MFS	72.7	70.1	62.4	69.0	72.0	70.5
BERT$_{GLU+LW}$	75.5	73.4	68.5	71.0	76.2	74.0
GlossBERT	77.7	75.2	**76.1**	72.5	80.4	77.0
LMMS	76.3	75.6	68.1	75.1	77.0	75.4
UKB$_{+Syn}$	71.2	71.6	59.6	72.4	75.6	71.5
SensEmBERT	70.8	65.4	58.0	74.8	75.0	70.1
SenseEmBERT$_{sup}$	72.2	69.9	60.2	**78.7**	75.0	72.8
ARES	78.0	<u>77.1</u>	71.0	77.3	**83.2**	77.9
Proposed Method						
CDES$_{linear}$	**78.4**	76.9	71.0	77.6	<u>83.1</u>	<u>78.0</u>
CDES$_{ReLU}$	<u>78.1</u>	<u>77.1</u>	71.0	77.5	<u>83.1</u>	<u>78.0</u>
CDES$_{GELU}$	<u>78.1</u>	**77.3**	<u>71.4</u>	<u>77.7</u>	**83.2**	**78.1**

Table 3.2: F1 scores (%) for English all-words WSD on the test sets of Raganato et al. (2017). Bold and underline indicate the best and the second best results, respectively. The results obtained using CDES$_{GELU}$ are statistically significant compared to ARES (cf. paired t-test with $p < 0.05$).

meanings, it only consists of nouns due to the limitation of its underlying resources. SensEmBERT$_{sup}$ is the supervised version of SensEmBERT. ARES (Scarlini et al., 2020b) is a semi-supervised approach for learning sense embeddings by incorporating sense annotated datasets, unlabelled corpora and knowledge bases.

To study the effect of using a nonlinear mapping f between static and contextualised embedding spaces in (3.2), we train CDES with linear, ReLU and GELU activations to create respectively CDES$_{linear}$, CDES$_{ReLU}$ and CDES$_{GELU}$ versions. From Table 3.2 we see that among these versions, CDES$_{GELU}$ outperforms the linear and ReLU versions

in all datasets, except on SE2 where CDES$_{linear}$ performs best. This result shows that nonlinear mapping (GELU) to be more appropriate for extracting sense-related information from contextualised embeddings. Moreover, we see that CDES versions consistently outperform all previously proposed sense embeddings, except on SE07 and SE13 where GlossBERT and SenseBERT$_{sup}$ perform best respectively. On SE15, the performance of CDES$_{GELU}$ is equal to that of ARES. However, CDES versions do not surpass GlossBERT and SenseEmBERT$_{sup}$ on SE07 and SE13 datasets, respectively. Recall that both SE07 and SE13 have fewer instances compared to the other datasets. Specifically, SE07 does not contain adjectives and adverbs, while SE13 does not contain verbs, adjectives and adverbs as shown in Table 3.1.

Overall, CDES$_{linear}$ obtains the best performance on SE2, while CDES$_{GELU}$ performs best on SE3, SE15 and ALL. This result provides empirical support to our working hypothesis that contextualised embeddings produced by NLMs encode much more information beyond sense related information, which must be filtered out using $\mathbf{W}$. CDES is able to accurately extract the sense-specific information from contextualised embeddings generated by a pretrained NLM to create sense-specific versions of pretrained sense-agnostic static embeddings.

3.3.3 Words in Context (WiC)

Pilehvar and Camacho-Collados (2019) introduced the WiC dataset for evaluating sense embedding methods. For a particular word u, WiC contains pairs of sentences, (t_1, t_2) where the same (*positive*) or different (*negative*) senses of u can occur. An accurate sense embedding method must be able to discriminate the different senses of an ambiguous word. The problem is formalised as a binary classification task and classification accuracy is reported as the evaluation metric. A method that assigns the same vector to all of the senses of a word would report

a chance-level (i.e. 50%) accuracy on WiC.

Similar to § 3.3.2, we first determine the sense-specific embeddings of u, $\boldsymbol{s}_i(u)$ and $\boldsymbol{s}_j(u)$ for the senses of u used in respectively t_1 and t_2. We then train a binary logistic regression classifier using the train split of WiC, where we use the cosine similarities between the two vectors in the following six pairs as features, comparing sense and contextualised embeddings in the two sentences:

1. $(\boldsymbol{s}_i(u),\ \boldsymbol{s}_j(u))$: similarity between the sense embeddings of u in sentences t_1 and t_2, respectively.

2. $(\boldsymbol{\zeta}(u,t_1),\ \boldsymbol{\zeta}(u,t_2))$: similarity between the contextualised embeddings of u in sentences t_1 and t_2, respectively.

3. $(\boldsymbol{s}_i(u),\ \boldsymbol{\zeta}(u,t_1))$: similarity between the sense embedding of u and its contextualised embedding in sentence t_1.

4. $(\boldsymbol{s}_j(u),\ \boldsymbol{\zeta}(u,t_2))$: similarity between the sense embedding of u and its contextualised embedding in sentence t_2.

5. $(\boldsymbol{s}_i(u),\ \boldsymbol{\zeta}(u,t_2))$: similarity between the sense embedding of u in sentence t_1 and its contextualised embedding in sentence t_2.

6. $(\boldsymbol{s}_j(u),\ \boldsymbol{\zeta}(u,t_1))$: similarity between the sense embedding of u in sentence t_2 and its contextualised embedding in sentence t_1.

We train this classifier using the official train split in WiC. In particular, we do not fine-tune the static or contextualised embeddings that are used as inputs by CDES on WiC because our goal is to extract sense-related information already present in the pretrained embeddings.

In Table 3.3, we report the classification accuracies on WiC for different types of embeddings such as static word embeddings (GloVe), contextualised embeddings generated by NLMs (ELMo, ElMo-weighted, BERT-large, RoBERTa and KnowBERT), and sense-specific embeddings (MUSE, LMMS, LessLex, SenseBERT-large and BERT$_{ARES}$).

Models	Accuracy %
Static Embeddings	
GloVe (Pennington et al., 2014)	50.9
Contextualised Embeddings	
ElMo (Peters et al., 2018)	57.7
ELMo-weighted (Ansell et al., 2019)	61.2
BERT-large (Devlin et al., 2019)	65.5
RoBERTa (Liu et al., 2019)	69.9
KnowBERT-W+W (Peters et al., 2019a)	70.9
SenseBERT-large (Levine et al., 2020)	<u>72.1</u>
BERT$_{ARES}$ (Scarlini et al., 2020b)	**72.2**
Static Sense Embeddings	
MUSE (Lee and Chen, 2017)	48.4
LMMS (Loureiro and Jorge, 2019b)	67.7
LessLex (Colla et al., 2020)	59.2
CDES$_{linear}$	69.0
CDES$_{ReLu}$	68.6
CDES$_{GELU}$	68.8

Table 3.3: Performance on WiC. Bold and underline respectively indicate the best and the second best results.

From Table 3.3 we see that SenseBERT-large and BERT$_{ARES}$ obtain better performance than other embeddings. All the CDES variants outperform previous static sense embeddings learning methods. However, MUSE[5] does not assign sense labels to sense embeddings as done by LMMS, LessLex and CDES. Among CDES variants, CDES$_{linear}$ performs best and is closely followed by GELU and ReLU variants. Although, CDES variants do not surpass the current SoTA methods such as SenseBERT-large and BERT$_{ARES}$ on WiC, unlike CDES these methods fine-tune on WiC train data and/or use more complex classifiers with multiple projection layers compared to the single logistic

[5] https://github.com/MiuLab/MUSE

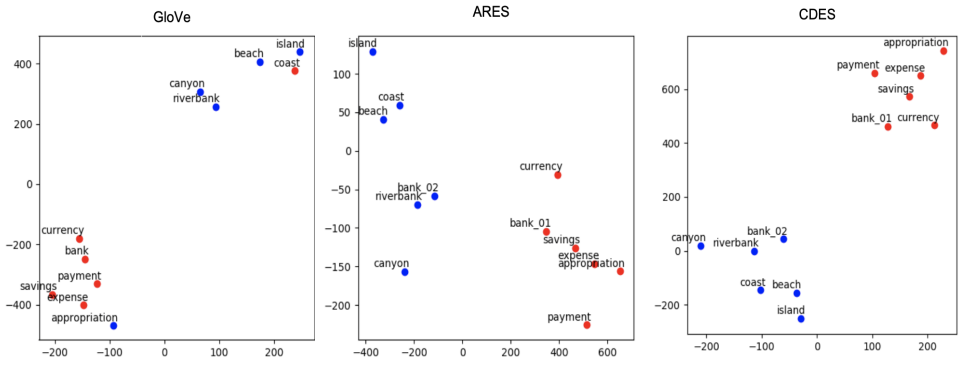


Figure 3.2: t-SNE visualisations of the nearest neighbours of *bank* corresponding to the two senses *financial institution* (in red) and *sloping land* (in blue) are shown for GloVe, ARES and CDES embeddings. Sense labels of synonyms are omitted to avoid cluttering.

regression over six features used by CDES.[6] More importantly, results from both WSD and WiC experiments support our claim that contextualised embeddings encode word sense related information that can be extracted and injected into sense-insensitive static word embeddings via (non)linear projections to create sense-sensitive versions of the sense-insensitive static embeddings.

3.3.4 Visualisation of Sense Embeddings

To visualise the embeddings corresponding to the different senses of an ambiguous word, we consider *bank*, which has the two distinct senses according to WordNet: *financial institution (sense-key=bank%1:14:00::)* and *sloping land (sense-key=bank%1:17:01::)*. We randomly select 5 synonyms for each sense of *bank* from the WordNet and project their sense/word embeddings using t-SNE (Van der Maaten and Hinton, 2008) in Figure 3.2. Compared to GloVe, we see that words with related

[6] BERT_{ARES} and SenseBERT use respectively 2048 and 1024 features for sense prediction in WiC.

Sentence 1: The banks which held the mortgage on the old church declared that the interest was considerably in arrears, and the real estate people said flatly that the land across the river was being held for an eventual development for white working people who were coming in, and that none would be sold to colored folk.				
GloVe	BERT	LMMS	SenseBERT	CDES
mortgage	mortgage	mortgage	mortgage	mortgage
interest	interest	church	real	real estate
estate	held	sell	old	sell
river	church	interest	land	interest
real	river	real estate	interest	church
Sentence 2: Through the splash of the rising waters, they could hear the roar of the river as it raged through its canyon, gnashing big chunks out of the banks.				
GloVe	BERT	LMMS	SenseBERT	CDES
mortgage	river	river	splash	river
interest	waters	canyon	land	water
estate	chunks	land	out	rise
river	splash	folk	through	canyon
real	canyon	church	chunks	folk

Table 3.4: Nearest neighbours computed using the word/sense embeddings of *bank* in two sentences.

meanings are projected onto coherent clusters by ARES and CDES. This indicates that sense embeddings are able to distinguish polysemy correctly compared to static word embeddings. Overall, we see that CDES produces better separated clusters than both GloVe and ARES.

3.3.5 Nearest Neighbours of Sense Embeddings

An accurate sense embedding method must be able to represent an ambiguous word with different embeddings considering the senses ex-

pressed by that word in different contexts. To understand how the sense embedding of a word varies in different contexts, we compute the nearest neighbours of an ambiguous word using its sense embedding. Table 3.4 shows two sentences from SemCor containing *bank*, where in Sentence 1, *bank* takes the *financial institution* sense, and in Sentence 2 the *sloping land (especially the slope beside a body of water)* sense. We compute the sense embedding of *bank*, given each sentence as the context, using different methods and compute the top 5 nearest neighbours, shown in the descending order of their cosine similarity scores with the sense embedding of *bank* in each sentence.

GloVe, which is sense and context insensitive uses the same vector to represent *bank* in both sentences, resulting in the same set of nearest neighbours, which is a mixture of finance and riverbank related words. On the other hand, BERT, which is context-sensitive but not sense-specific, returns different sets of nearest neighbours in the two cases. In particular, we see that finance-related nearest neighbours such as *mortgage* and *interest* are selected for the first sentence, whereas riverbank-related nearest neighbours such as *water* and *canyon* for the second. However, BERT does not provide sense embeddings and some neighbours such as *river* appear in both sets, because it appears in the first sentence, although not related to *bank* there.

SenseBERT (Levine et al., 2020) disambiguates word senses at a coarse-grained WordNet's supersense level. We see that SenseBERT correctly detects words such as *mortgage* and *interest* as neighbours of *bank* in the first sentence, and *splash* and *land* in the second. We see that *land* appears as a nearest neighbour in both sentences, although it is more related to the *sloping land* sense than the *financial institution* sense of *bank*.

LMMS selects *church* as the nearest neighbour for both sentences, despite being irrelevant to the second. On the other hand, CDES correctly detects *church* for the first sentence and not for the second. Over-

all, CDES correctly lists financial institution sense related words such as *mortgage*, *real estate* and *interest* for the first sentence, and sloping land sense related words such as *river*, *water* and *canyon* in the second sentence.

3.4 Summary

In this chapter, we introduce CDES, a method which is able to generate sense embeddings by extracting the sense-related information from contextualised embeddings. CDES integrates the gloss information from a semantic network as well as the information from an external corpus to tackle the sense-coverage problem. Evaluations on multiple benchmark datasets related to WSD and WiC tasks show that CDES learns accurate sense embeddings, and report comparable results to the current SoTA. All experiments reported in the thesis are limited to the English language and we plan to extend the proposed method to learn multilingual sense embeddings in our future work.

Since different methods have been proposed in prior work on sense embedding learning that use different sense inventories, sense-tagged corpora and learning methods. However, not all existing sense embeddings cover all senses of ambiguous words equally well due to the discrepancies in their training resources. We will address this problem in the next chapter.

4 Meta Sense Embeddings Learning

4.1 Introduction

Prior work has shown that sense embeddings are useful for tasks such as WSD and sense discrimination tasks such as WiC (Loureiro and Jorge, 2019b; Pilehvar and Camacho-Collados, 2019). However, existing sense embeddings are trained on diverse resources such as sense tagged corpora or dictionary glosses, with varying levels of sense coverage (e.g. fully covering all synsets in the WordNet vs. a subset), and using different methods. Therefore, the performance reported by the existing sense embeddings on different downstream tasks and datasets varies significantly for different PoS categories. Moreover, it is not readily clear which sense embedding learning method should be used for disambiguating words in a given domain.

To address these problems, we propose a method that incorporates multiple independently pretrained **source** sense embeddings to learn a **meta**-sense embedding such that the sense-related information captured by the source (input) sense embeddings is preserved in the (output) meta-sense embedding. Our proposed method can combine full-coverage sense embeddings with partial-coverage ones, thereby improving the sense coverage in partial-coverage sense embeddings. We project

each source sense embedding space into a common meta-sense embedding space using source-specific projection matrices. The meta-sense embedding is then computed as the average of the projected source sense embeddings in this meta-embedding space.

The meta-sense embedding $\boldsymbol{m}(s)$ of a sense s is required to satisfy two distinct criteria:

1. $\boldsymbol{m}(s)$ is required to be similar to all of the source embeddings of s. This preserves the sense-related information from the individual source embeddings in their meta-sense embedding. We use PIP to compare the similarity distributions (nearest neighbours) over senses between meta and source embedding spaces.

2. $\boldsymbol{m}(s)$ is required to be similar to the contextualised embeddings of the contexts in which s occurs. This ensures that meta-sense embeddings could be used in downstream tasks such as WSD or WiC, where we must select the correct sense of an ambiguous word given its context.

The sense-specific projection matrices are learned such that both the above-mentioned criteria are simultaneously satisfied. We name our proposed method Neighbourhood Preserving Meta-Sense Embedding (NPMS).

Meta-embedding learning has been successfully used to learn word-level and sentence-level meta-embeddings in prior work (Bollegala and O'Neill, 2022; Yin and Schütze, 2016). However, to the best of our knowledge, meta-embedding learning methods have *not* been applied for sense embeddings before. Indeed, compared to word-level meta-embedding, sense-level meta-embedding has several unique challenges, which are addressed by NPMS.

Challenge 1 (*missing senses*). Compared to learning meta-word embeddings, where each word is assigned a single embedding, in static

sense embeddings an ambiguous word is associated with multiple sense embeddings, each corresponding to a distinct sense of the ambiguous word. However, not all of the different senses of a word might be equally covered by all source sense embeddings. NPMS does *not* compare the source embeddings directly but requires the nearest neighbours computed using source and meta sense embeddings to be similar. This allows us to use shared neighbours to compute the alignment between source and meta-embedding spaces, without having to predict the missing sense embeddings.

Challenge 2 (*Misalignment between sense and context embeddings*). In downstream tasks such as WSD, we must determine the correct sense s of an ambiguous word w in a given context (i.e. a sentence) c. This is done by comparing the sense embeddings for each distinct sense of w against the context embedding of c, for example, computed using a MLM such as BERT. The sense corresponding to the sense embedding that has the maximum similarity with the context embedding is then selected as the correct sense of w in c. For sense embeddings such as LMMS or ARES this is trivially achieved because they are both BERT-based embeddings and the cosine similarity between those sense embeddings and BERT embeddings can be directly computed. However, this is *not* the case for meta-sense embeddings that exist in a different vector space than the context embeddings produced by BERT. Therefore, we must first learn a projection between the meta-sense and the context embedding spaces when conducting WSD. On the other hand, NPMS embeddings are learned such that they can be directly compared with BERT-based contextualised embeddings as-is, without requiring any costly projections.

We evaluate NPMS on WiC and WSD datasets and compare against several competitive baselines for meta-embedding learning. Our experimental results show that NPMS consistently outperforms all other

methods in both tasks. More importantly, we obtain SoTA performance for WSD and WiC reported by a static sense embedding method.

4.2 Meta-Sense Embedding Learning

To explain our proposed method in detail, let us first consider a vocabulary $\mathcal{V}$ of words $w \in \mathcal{V}$. We further assume that each word w is typically associated with one or more distinct senses s and the set of senses associated with w is denoted by $\mathcal{S}_w$. In meta-sense embedding learning, we assume a sense s of a word to be represented by a set of n source sense embeddings. Let us denote the j-th source embedding of s by $\boldsymbol{x}_j(s) \in \mathbb{R}^{d_j}$, where d_j is the dimensionality of the j-th source embedding.

We project the j-th source embedding by a matrix $\mathbf{P}_j \in \mathbb{R}^{d_j \times d}$ into a common meta-sense embedding space with dimensionality d. The meta-sense embedding, $\boldsymbol{m}(s) \in \mathbb{R}^d$ of s is computed as the unweighted average of the projected source sense embeddings as given by (4.1).

$$\boldsymbol{m}(s) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{P}_j^{\intercal} \boldsymbol{x}_j(s) \tag{4.1}$$

After this projection step, all source sense embeddings live in the same d-dimensional vector space, thus enabling us to add them as done in (4.1). An advantage of considering the average of the projected source embeddings as the meta-sense embedding is that, even if a particular sense is not covered by one or more source sense embeddings, we can still compute a meta-sense embedding using the remainder of the source sense embeddings. Moreover, prior work on word-level and sentence-level meta-embedding has shown that averaging after a linear projection to improve performance when learning meta embeddings (Coates and Bollegala, 2018; Jawanpuria et al., 2020; Poerner et al., 2020).

If we limit the projection matrices to be orthonormal, they can be seen as optimally rotating the source sense embeddings such that the projected source embeddings could be averaged in the meta-embedding space. However, we observed that dropping this regularisation term produces better meta-sense embeddings in our experiments. Therefore, we did not impose any orthonormality restrictions on the projection matrices.

We require a meta-sense embedding to satisfy two criteria: (a) **sense information preservation** and (b) **contextual alignment**. The two criteria jointly ensure that the meta-sense embeddings we learn are accurate and can be used in downstream tasks such as WSD in conjunction with contextualised word embeddings such as BERT. Next, we describe each of those criteria in detail.

4.2.1 Sense Information Preservation

Given that the individual source sense embeddings are trained on diverse sense-related information sources, we would like to preserve this information as much as possible in the meta-sense embeddings we create from those source sense embeddings. This is particularly important in meta-embedding learning because we might not have access to all the resources that were used to train the individual source sense embeddings, nor we will be training meta-embeddings from scratch but will be relying upon pretrained sense embeddings as the sole source of sense-related information into the meta-embedding learning process. Therefore, we must preserve the complementary sense-related information encoded in the source sense embeddings as much as possible in their meta-sense embedding.

It is not possible however to directly compare the meta-sense embeddings computed using (4.1) against the source sense embeddings because they have different dimensionalities and live in different vector spaces.

This makes it challenging when quantifying the amount of information lost due to meta embedding using popular loss functions such as squared Euclidean distance between source and meta embeddings. To address this problem we resort to PIP, which has been previously used to determine the optimal dimensionality of word embeddings (Yin and Shen, 2018) and learning concatenated word-level meta embeddings (Bollegala, 2022).

Given a source/meta embedding matrix $\mathbf{E}$, the corresponding PIP matrix is given by (4.2)

$$\text{PIP}(\mathbf{E}) = \mathbf{E}\mathbf{E}^{\top} \tag{4.2}$$

Specifically, PIP matrix contains the inner products between all pairs of sense embeddings represented by the rows of $\mathbf{E}$. $\text{PIP}(\mathbf{E})$ is a symmetric matrix with its number of rows (columns) equal to the total number of unique senses covering all the words in the vocabulary.

Let us denote the source sense embedding matrix for the j-th source by $\mathbf{X}_j$, where the i-th row represents sense embedding $\boldsymbol{x}_j(s_i)$ learned for the i-th sense s_i. Likewise, let us denote by $\mathbf{M}$ the meta-sense embedding matrix, where the i-th row represents the meta-sense embedding $\boldsymbol{m}(s_i)$ computed for s_i using (4.1). Because the shape of PIP matrices are independent of the dimensionalities of the embedding spaces, and the rows are aligned (i.e. sorted by the sense ids s_i), we can compare the meta-sense embedding against the individual source sense embedding using PIP loss, L_{pip}, given by (4.3).

$$L_{\text{pip}} = \sum_{j=1}^{n} ||\text{PIP}(\mathbf{X}_j) - \text{PIP}(\mathbf{M})||_F^2 \tag{4.3}$$

Here, $||\mathbf{A}||_F = \sqrt{\sum_{l,m} a_{lm}^2}$ denotes the Frobenius norm of the matrix $\mathbf{A}$. PIP loss can be seen as comparing the distributions of similarity

scores computed using the meta-sense embedding and each of the individual source sense embeddings for the same set of senses. Although the actual vector spaces might be different and initially not well-aligned due to the projection and averaging steps in (4.1), we would require the neighbourhoods computed for each word to be approximately similar in the meta-sense embedding space and each of the source sense embedding spaces. PIP loss given in (4.3) measures this level of agreement between meta and source embedding spaces.

4.2.2 Contextual Alignment

As described in Chapter 3, the context in which an ambiguous word has been used provides useful clues to determine the correct sense of that word. For example, consider the following two sentences: **(S1)** *I went to the bank to withdrew some cash.*, and **(S2)** *The river bank was crowded with people doing BBQs.* Words *cash* and *withdrew* indicate that it is the *financial institute* sense of the bank appearing in **S1**, whereas the words *river*, *BBQ* indicate that it is the *sloping land* sense of bank appearing in **S2**.

Let us denote the contextualised word embedding of a word w in a context c by $\boldsymbol{f}(w; c)$. MLMs such as BERT and RoBERTa have been used in prior work in WSD to compute context-sensitive representations for ambiguous words. Then, the above-described agreement between the sense s of w and its context c can be measured by the similarity between the meta-sense embedding $\boldsymbol{m}(s)$ and the contextualised embedding $\boldsymbol{f}(w; c)$. We refer to this requirement as the *contextual alignment* between a meta-sense embedding and contextualised word embeddings.

Given a sense annotated dataset such as SemCor, we represent it by a set $\mathcal{T}$ of tuples (w, s, c), where the word w is annotated with its correct sense s in context c. Then, we define the contextual alignment loss L_{cont} as (negative) average cosine similarity between $\boldsymbol{m}(s)$ and $\boldsymbol{f}(w; c)$, given

by (4.4).

$$L_{\text{cont}} = -\sum_{(w,s,c)\in\mathcal{T}} \frac{\boldsymbol{m}(s)^{\top}\boldsymbol{f}(w;c)}{||\boldsymbol{m}(s)||_2\,||\boldsymbol{f}(w;c)||_2} \tag{4.4}$$

Minimising the contextual alignment loss in (4.4), will maximise the cosine similarity between the meta-sense embedding and the corresponding contextualised embedding.

In contrast to the PIP-loss defined by (4.3), which can be computed without requiring sense annotated data, the contextual alignment loss defined by (4.4) requires sense annotated data. However, SemCor, the sense annotated dataset that we use for computing the contextual alignment loss in this thesis, is already being used by many existing pretrained source sense embeddings. Therefore, we emphasise that we are *not* requesting any additional training resources during the meta-sense embedding learning process beyond what has been already used to train the source sense embeddings. Moreover, ablation studies (§ 4.3.4) show that PIP-loss alone obtains significant improvements, without the contextual alignment loss.

Contextual alignment loss can also be motivated from an application perspective. Sense embeddings are often used to represent word senses in downstream tasks such as WSD. A typical approach for predicting the sense of an ambiguous word w as used in a given context c is to measure the cosine similarity between each sense embedding of w and the context embedding for c (Loureiro and Jorge, 2019a; Scarlini et al., 2020b). The objective given in (4.4) can be seen as enforcing this property directly into the meta-sense embedding learning process. As we later see in § 4.3, the meta-sense embeddings learned by NPMS perform particularly well in WSD benchmarks.

In order to compute the cosine similarity between meta-sense embeddings and contextualised word embeddings, we must first ensure that

they have the same dimensionality. This can be achieved by setting the dimensionality of the meta-sense embeddings equal to that of the contextualised word embeddings. Alternatively, we can learn a projection matrix that adjusts the dimensionality of the meta-sense embeddings to that of the contextualised word embeddings. However, in our preliminary investigations, we did not observe any significant performance gains despite the additional parameters introduced by the projection matrix that must be learned for this purpose. Therefore, in our experiments, we set the dimensionality of the meta-sense embeddings to that of the contextualised word embeddings.

4.2.3 Parameter Learning

We consider the linearly-weighted sum of the PIP-loss and contextual alignment loss as the total loss, L_{tot}, given by (4.5).

$$L_{\text{tot}}(\{\mathbf{P}_j\}_{j=1}^{n}) = \alpha L_{\text{pip}} + (1-\alpha) L_{\text{cont}} \tag{4.5}$$

Here, the parameters to be learned are the projection matrices $\mathbf{P}_j$ for the sources $j = 1, \ldots, n$. The weighting coefficient $\alpha \in [0, 1]$ determines the emphasis of the two losses, which is a hyperparameter. In our experiments, we tune α using a validation set of Senseval-3 WSD dataset (Snyder and Palmer, 2004).

Compared to the cosine similarity, which is upper bounded by 1, the PIP-loss grows with the size of the PIP matrices being used. Therefore, we found that scaling the two losses by their mean values is important to stabilise the training. We initialise the projection matrices to the identity matrix and use vanilla stochastic gradient descent with a learning rate of 0.001, determined using the validation set of the Senseval3 WSD dataset.

4.3 Experiments and Results

In this section, we evaluate the performance of our proposed NPMS with different settings on both WSD and WiC tasks.

4.3.1 Source Embeddings

Our proposed NPMS is agnostic to the methods used to learn the source sense embeddings, and thus in principle can be used to meta-embed any source sense embedding. In our experiments, we use the following source sense embeddings because of their state-of-the-art performance, public availability and coverage of WordNet senses.

LMMS_{2048} is a supervised approach to learning full-coverage static sense embeddings that cover all of the 206,949 senses in the WordNet. LMMS uses BERT, semantic networks (i.e., WordNet) and glosses to create sense embeddings with 2048 dimensions. They obtained a 2348 dimension sense embedding LMMS_{2348} by appending the static word embedding generated from fastText (Bojanowski et al., 2017) to increase robustness. As the performance of LMMS_{2348} and LMMS_{2048} are comparable, except for the Uninformed Sense Matching (USM) task (Loureiro and Jorge, 2019a), we selected LMMS_{2048}.

SenseEmBERT obviates the need for sense-annotated corpora by using the BabelNet[1] mappings between WordNet senses and Wikipedia pages to construct sense embeddings with 2048 dimensions, covering all the 146,312 English nominal senses of WordNet. Each sense embedding consists of two components: (a) the average of the word embedding of the target sense's relevant words, and (b) the average of the BERT

[1] babelnet.org

encoded tokens of the sense gloss. For the brevity of the notation, we denote SenseEmBERT as **SBERT** in the remainder of this thesis.

ARES is a semi-supervised method that learns sense embeddings with full coverage of the WordNet and is 2048 dimensional. ARES embeddings are created by applying BERT on the glossary information and the information contained in the SyntagNet (Maru et al., 2019). It outperforms LMMS in WSD benchmarks.

The intersection of the LMMS_{2048} and ARES contains 206,949 senses, which is equivalent to the total number of senses in the WordNet because they both cover all the senses in the WordNet (i.e. full coverage sense embeddings). On the other hand, the intersection between the LMMS_{2048} and SensEmBERT as well as the intersection between the ARES and SensEmBERT contains 146,312 senses, which is the total number of nominal senses in the WordNet. By using source sense embeddings with different sense coverages we aim to evaluate the ability of meta-sense embedding methods to learn accurate sense embeddings by exploiting the complementary strengths in the sources.

4.3.2 Evaluation Tasks

We compare the accuracy of meta-sense embeddings using two standard tasks that have been used in prior work on sense embedding learning.

Word Sense Disambiguation (WSD): To test whether NPMS can disambiguate the different senses of an ambiguous word, we conduct a WSD task using the evaluation framework proposed by Raganato et al. (2017) described in § 2.3.1.2. We use the framework's official scoring scripts to avoid any discrepancies in the scoring methodology.

Similar to § 3.3.2, we perform the WSD following the 1-NN procedure. Specifically, we compute the contextualised embedding, de-

noted by $\boldsymbol{f}(w; c)$, using BERT by averaging the last four layers for each word w in a test sentence c. We then measure the cosine similarity, $\phi(\boldsymbol{m}(s), \boldsymbol{f}(w; c))$, between the source/meta sense embedding for each sense s of w, $\boldsymbol{m}(s)$, and $\boldsymbol{f}(w; c)$, and select the sense that produces the maximum cosine similarity as the correct sense of w in c.

Word-in-Context (WiC): Given a target word w in two contexts c_1 and c_2, we first determine the meta-sense embeddings of w, which are $\boldsymbol{m}(s_1)$ and $\boldsymbol{m}(s_2)$ corresponding to the senses of w used in respectively c_1 and c_2. Let the BERT representation of w in both c_1 and c_2, as $\boldsymbol{f}(w; c_1)$ and $\boldsymbol{f}(w; c_2)$. We train a binary logistic regression classifier on the WiC training set. Similar to § 3.3.3, we use the cosine similarities between the two vectors in the following six pairs as features:

1. $\phi(\boldsymbol{m}(s_1), \boldsymbol{m}(s_2))$: similarity between the sense embeddings of w in sentences c_1 and c_2, respectively.

2. $\phi(\boldsymbol{f}(w; c_1), \boldsymbol{f}(w; c_2))$: similarity between the contextualised embeddings of w in sentences c_1 and c_2, respectively.

3. $\phi(\boldsymbol{m}(s_1), \boldsymbol{f}(w; c_1))$: similarity between the sense embedding of w and its contextualised embedding in sentence c_1.

4. $\phi(\boldsymbol{m}(s_2), \boldsymbol{f}(w; c_2))$: similarity between the sense embedding of w and its contextualised embedding in sentence c_2.

5. $\phi(\boldsymbol{m}(s_1), \boldsymbol{f}(w; c_2))$: similarity between the sense embedding of w in sentence c_1 and its contextualised embedding in sentence c_2.

6. $\phi(\boldsymbol{m}(s_2), \boldsymbol{f}(w; c_1))$: similarity between the sense embedding of w in sentence c_2 and its contextualised embedding in sentence c_1.

4.3.3 Meta-Embedding Methods

We extend prior works on meta-embedding learning to meta-sense embedding learning by taking the sense embeddings described in § 4.3.1 as source embeddings, and compare them with NPMS embeddings. **AVG** (Coates and Bollegala, 2018) takes the average over the embeddings of a sense from different sources embeddings. **CONC** (Yin and Schütze, 2016) creates meta-embeddings by concatenating the embeddings from different source embeddings. **SVD** (Yin and Schütze, 2016) performs dimensionality reduction on the concatenated source embeddings. **AEME** (Bollegala and Bao, 2018) is an autoencoder-based method for meta-embedding learning, which is the current state-of-the-art unsupervised word-level meta-embedding learning method. We use 2048 output dimensions for both SVD and AEME in the experiments, determined to be the best for those methods on validation data.

As noted in § 4.3.2, both WSD and WiC tasks require us to compute the cosine similarity, ϕ, between a source/meta sense embedding, $\boldsymbol{m}(s)$, of a sense s and a contextualised word embedding, $\boldsymbol{f}(w; c)$, of the ambiguous word w in context c. However, unlike for NPMS, which explicitly guarantees that its meta-sense embeddings are directly comparable with the contextualised word embeddings via the contextual loss (4.4), in general, the meta-sense embeddings produced by other methods do not exist in the BERT embedding space and require careful consideration as we discuss next.

Let us first consider computing ϕ between the source sense embeddings and BERT embeddings. All three source sense embeddings we use are 2048-dimensional and they are computed by concatenating two 1024-dimensional BERT embeddings, averaged over different lexical resources. Therefore, using the same 1024-dimensional BERT and by concatenating $\boldsymbol{f}(w; c)$ twice, we can obtain a 2048-dimensional BERT-based embedding for w that can be used to compute the cosine similarity

with a source sense embedding.

Next, let us consider the meta-sense embeddings produced by CONC. Because inner-product decomposes trivially over vector concatenation, we can copy and concatenate $\boldsymbol{f}(w;c)$ to match $\boldsymbol{m}(s)$ produced by CONC. For example, if CONC is used with LMMS and ARES, we can concatenate $\boldsymbol{f}(w;c)$ four times, and then compute the inner product with the meta-sense embedding. AVG does not change the dimensionality of the meta-sense embedding space. Therefore, we only need to concatenate $\boldsymbol{f}(w;c)$ twice when computing the cosine similarity with AVG for any number of source sense embeddings.

Unfortunately, the meta-sense embedding spaces produced by SVD and AEME are not directly comparable to that of BERT embeddings due to the differences in dimensionality and non-linear transformations introduced (cf. AEME uses autoencoders). Therefore, we learn a projection matrix, $\mathbf{A}$, between $\boldsymbol{m}(s)$ and $\boldsymbol{f}(w;c)$ by minimising the squared Euclidean distance given by (4.6), computed using the SemCor training dataset, $\mathcal{T}$.

$$\sum_{(w,s,c)\in\mathcal{T}} ||\mathbf{A}\boldsymbol{m}(s) - \boldsymbol{f}(w;c)||_2^2 \tag{4.6}$$

After training, we compute the cosine similarity, $\phi(\mathbf{A}\boldsymbol{m}(s), \boldsymbol{f}(w;c))$, between the transformed SVD and AEME meta-sense embedding and BERT embeddings.

4.3.4 Results

Table 4.1 compares the performance of NPMS against the meta-embedding methods described in § 4.3.3 on WSD and WiC. We see that NPMS obtains the overall best performance for WSD (ALL) as well as on the WiC. Among the three sources, ARES reports the best performance for WSD (ALL), while SBERT does so for WiC. In SE2, SE07 datasets

	SE2	SE3	SE07	SE13	SE15	ALL	WiC
LMMS	76.34	75.57	68.13	75.12	77.01	75.44	69.30
ARES	78.05	77.08	70.99	77.31	**83.17**	77.91	68.50
SBERT	53.11	52.22	41.37	**78.77**	55.12	59.85	71.14
AVG	79.36	**77.46**	70.33	77.86	80.82	78.17	71.16
CONC	78.22	77.14	70.99	77.37	82.97	77.97	70.38
SVD	75.02	74.22	67.25	72.81	74.85	73.80	63.01
AEME	78.53	76.92	69.01	76.09	78.96	77.03	70.69
NPMS	**79.93**	77.30	**71.65**	77.49	81.21	**78.37**	**71.47**

Table 4.1: F1 scores on WSD benchmarks and accuracy on WiC are shown for the three sources (top) and for the different meta-embedding methods (bottom).

NPMS report the best performance, whereas AVG, SBERT and ARES do so respectively in SE3, SE13 and SE15. Among the baseline methods, we see AVG reports the best results, which is closely followed by CONC. Poor performance of SVD shows the challenge of applying dimensionality reduction methods on CONC due to missing sense embeddings. Although AEME has reported the SoTA performance for word-level meta-embedding, applying it directly on sense embeddings is suboptimal. This shows the difference between word- vs. sense-level meta-embedding learning problems and calls for sense-specific meta-embedding learning methods.

According to the WiC leader board[2], the performance reported by NPMS is second only to SenseBERT, which is a contextualised sense embedding method obtained by fine-tuning BERT on WordNet supersenses. Therefore, the performance of NPMS can be seen as the SoTA for any *static* sense embedding method.

To further study the effect of using different sources, we compare the meta-embeddings produced for all pairwise combinations of sources

[2] https://pilehvar.github.io/wic/

	SE2	SE3	SE07	SE13	SE15	ALL	WiC
				ARES+LMMS			
AVG	**78.79**	77.03	69.89	77.13	**81.80**	77.83	70.22
CONC	78.75	76.76	70.33	77.31	81.12	77.72	**70.53**
SVD	76.30	74.20	68.40	74.00	76.30	74.80	66.93
AEME	77.39	75.46	67.25	75.85	78.18	76.02	68.65
NPMS	78.53	**77.14**	**71.87**	**77.37**	81.60	**77.93**	70.22
				ARES+SBERT			
AVG	78.57	77.35	71.21	78.10	**81.70**	78.13	71.32
CONC	78.79	**77.68**	71.21	78.41	81.21	78.28	**71.47**
SVD	75.24	73.68	65.71	72.69	75.73	73.74	66.46
AEME	77.78	75.41	69.23	76.22	78.77	76.42	68.18
NPMS	**78.79**	77.41	**71.65**	**78.53**	81.41	**78.30**	71.32
				LMMS+SBERT			
AVG	77.70	76.16	68.79	78.04	77.69	76.82	69.59
CONC	77.39	76.27	69.23	78.10	77.79	76.81	71.00
SVD	74.67	73.62	66.15	71.84	75.93	73.40	66.30
AEME	75.99	75.03	64.40	75.79	77.01	75.11	68.34
NPMS	**78.05**	**76.86**	**69.89**	**78.28**	**78.28**	**77.32**	**71.79**

Table 4.2: F1 scores on WSD benchmarks and accuracy on WiC are shown for the meta-embeddings created from all pairwise combinations of source embeddings.

in Table 4.2. We see that NPMS consistently outperforms other methods on the WSD (ALL) dataset, which indicates the effectiveness of the meta-sense embeddings learned using NPMS on word sense disambiguation. We see that ARES+SBERT and LMMS+SBERT are the best source combinations for NPMS respectively in WSD and WiC. This is particularly interesting because, unlike ARES and LMMS, SBERT does not cover all the senses in the WordNet but can obtain superior results in both WSD and WiC tasks simply by meta-embedding with a full-coverage sense embedding such as ARES or LMMS. This is attractive from an application perspective because it shows that NPMS

Method	WSD (ALL)	WiC
SVD with proj.	74.80	66.93
SVD without proj.	35.90	60.34
AEME with proj.	76.02	68.65
AEME without proj.	41.60	53.61

Table 4.3: Effect of learning a projection matrix between meta-sense vs. BERT embedding spaces.

is an effective technique to increase the coverage of a pretrained sense embedding.

Table 4.3 shows the importance of learning a projection matrix via (4.6) between meta-sense and BERT embeddings, for SVD and AEME. We see that the performance of both of those methods drops significantly without the projection matrix learning step. Even with projection matrices, SVD and AEME do not outperform simpler baselines such as AVG or CONC. On the other hand, NPMS does not require such a projection matrix learning step and consistently outperforms all those methods across multiple WSD and WiC benchmarks.

To understand the contributions of the two loss terms PIP-loss (L_{pip}) and contextual alignment loss (L_{cont}), we conduct an ablation study where we train NPMS with three sources using only one of the two losses at a time. From Table 4.4, we see that in both WiC and WSD (ALL, SE2, SE3, SE15), the best performance is obtained by using both losses. Each loss contributes differently in different datasets, although the overall difference between the two losses is non-significant (cf. paired t-test with $p < 0.05$). This is particularly encouraging because PIP-loss can be computed without having access to a sense labeled corpus such as SemCor. Such resources might not be available in specialised domains such as medicine or e-commerce. Therefore, in such cases, we can still apply NPMS trained using only the PIP-loss. Although we considered a linearly-weighted combination of the two losses in (4.5),

	SE2	SE3	SE07	SE13	SE15	ALL	WiC
Both	**79.93**	**77.30**	71.65	77.49	**81.21**	**78.37**	**71.47**
L_{pip} only	79.80	77.03	**71.87**	77.49	80.72	78.20	70.69
L_{cont} only	79.54	77.19	70.77	**77.86**	80.33	78.12	71.32

Table 4.4: Ablation between the PIP-loss (L_{pip}) and contextual alignment loss (L_{cont}).

	SE2	SE3	SE07	SE13	SE15	ALL
AVG	78.83	**77.30**	70.55	77.07	82.00	77.97
CONC	78.92	76.92	70.77	**77.31**	81.12	77.84
SVD	75.50	72.22	65.71	72.26	75.05	73.25
NPMS	**79.10**	76.86	**71.65**	77.19	**82.58**	**78.02**

Table 4.5: Meta-embedding of 2348-dimensional LMMS and 2048-dimensional ARES source embeddings.

we believe further improvements might be possible by exploring more complex (nonlinear) combinations of the two losses.

In Table 4.5, we study meta-embeddings of different dimensional sources, where we use LMMS and ARES with dimensionalities respectively of 2348 and 2048. ARES embeddings are appropriately zero-padded when computing 2348-dimensional AVG meta-sense embeddings with LMMS. SVD reduces the dimensionality from 4396 (i.e. 2348 + 2048) to 2048, which is also the output dimensionality for NPMS. We see that NPMS is the overall best method, reporting the highest F1 in 3 out of the 5 WSD benchmarks. This result shows that our proposed NPMS can create meta-embeddings even with sources of different dimensionalities.

4.4 Summary

We proposed the first-ever meta-sense embedding learning method NPMS. For this purpose, we proposed two training criteria that must be simultaneously satisfied by a meta-sense embedding. Our proposed method can combine source sense embeddings that cover different sets of word senses. Experimental results on WiC and WSD datasets show that our proposed NPMS surpasses previously published results for static sense embedding, and outperforms multiple word-level meta-embedding learning methods when applied to sense embeddings.

One sense of an ambiguous word might be socially biased while its other senses remain unbiased. In comparison to the numerous prior work evaluating the social biases in pretrained word embeddings, the biases in sense embeddings have been relatively understudied. In the next chapter, we will study social biases in sense embeddings.

5 Social Biases in Senses Embeddings

5.1 Introduction

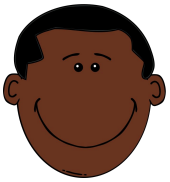
Although numerous prior works have studied social biases in static and contextualised word embeddings, social biases in sense embeddings remain underexplored (Kaneko and Bollegala, 2019, 2021, 2022b; Ravfogel et al., 2020; Dev et al., 2020; Schick et al., 2021; Wang et al., 2020). We follow Shah et al. (2020) and define social biases to be *predictive biases with respect to protected attributes* made by NLP systems. Even if a word embedding is unbiased, some of its senses could still be associated with unfair social biases. For example, consider the ambiguous word *black*, which has two adjectival senses according to the WordNet (Fellbaum and Miller, 1998): (1) black as a *colour* (*being of the achromatic colour of maximum darkness*, sense-key=**black%3:00:01**) and (2) black as a *race* (*of or belonging to a racial group especially of sub-Saharan African origin*, sense-key=**black%3:00:02**). However, only the second sense of *black* is often associated with racial biases.

Owing to (a) the lack of evaluation benchmarks for the social biases in sense embeddings, and (b) not being clear how to extend the bias evaluation methods that are proposed for static and contextualised em-

colour sense

Black dress is elegant.

Black dress is ugly.



race sense

Black people are friendly.

Black people are unfriendly.

Figure 5.1: Example sentences from the Sense-Sensitive Social Bias dataset for the two senses of the ambiguous word *black*. The top two sentences correspond to the colour sense of black, whereas the bottom two sentences correspond to its racial sense. Stereotypical examples that associate a sense with an unpleasant attribute are shown in red, whereas anti-stereotypical examples that associate a sense with a pleasant attribute are shown in blue.

beddings to evaluate social biases in sense embeddings, existing social bias evaluation datasets and metrics do not consider multiple senses of words, thus not suitable for evaluating biases in sense embeddings.

To address this gap, we evaluate social biases in SoTA static sense embeddings such as LMMS and ARES, as well as contextualised sense embeddings obtained from SenseBERT. To the best of our knowledge, we are the first to conduct a systematic evaluation of social biases in sense embeddings. Specifically, we make two main contributions to this thesis:

- First, to evaluate social biases in static sense embeddings, we extend previously proposed benchmarks for evaluating social biases in static (sense-insensitive) word embeddings by manually assign-

ing sense ids to the words considering their social bias types expressed in those datasets (§ 5.4).

- Second, to evaluate social biases in sense-sensitive contextualised embeddings, we create the Sense-Sensitive Social Bias (SSSB) dataset, a novel template-based dataset containing sentences annotated for multiple senses of an ambiguous word considering its stereotypical social biases (§ 5.6). An example from the SSSB dataset is shown in Figure 5.1.

Our experiments show that similar to word embeddings, both static as well as contextualised sense embeddings also encode worrying levels of social biases. Using SSSB, we show that the proposed bias evaluation measures for sense embeddings capture different types of social biases encoded in existing SoTA sense embeddings. More importantly, we see that even when social biases cannot be observed at the word level, such biases are still prominent at the sense level, raising concerns about existing evaluations that consider only word-level social biases.

5.2 Biases in Static Embedding

Our focus in this thesis is the evaluation of social biases in English and *not* the debiasing methods. We defer the analysis for languages other than English and develop debiasing methods for sense embeddings for future work. Hence, we limit the discussion here only to bias evaluation methods.

The Word Embedding Association Test (**WEAT**; Caliskan et al., 2017b) evaluates the association between two sets of target concepts (e.g. *male* vs. *female*) and two sets of attributes (e.g. Pleasant (*love, cheer*, etc.) vs. Unpleasant (*ugly, evil*, etc.)). Here, the association is measured using the cosine similarity between the word embeddings.

Ethayarajh et al. (2019) showed that WEAT systematically overestimates the social biases and proposed relational inner-product association (**RIPA**), a subspace projection method, to overcome this problem.

Word Association Test (**WAT**; Du et al., 2019) calculates a gender information vector for each word in an association graph (Deyne et al., 2019) by propagating information related to masculine and feminine words. Additionally, word analogies are used to evaluate gender bias in static embeddings (Bolukbasi et al., 2016; Manzini et al., 2019; Zhao et al., 2018b). Loureiro and Jorge (2019a) showed specific examples of gender bias in static sense embeddings. However, these datasets do not consider word senses and hence are unfit for evaluating social biases in sense embeddings.

5.3 Biases in Contextualised Embedding

May et al. (2019) extended WEAT to sentence encoders by creating artificial sentences using templates and using cosine similarity between the sentence embeddings as the association metric. Kurita et al. (2019) proposed the log-odds of the target and prior probabilities of the sentences computed by masking respectively only the target vs. both target and attribute words. Template-based approaches for generating example sentences for evaluating social biases do not require human annotators to write examples, which is often slow, costly and requires careful curation efforts. However, the number of sentence patterns that can be covered via templates is often small and less diverse compared to manually written example sentences.

To address this drawback, Nadeem et al. (**StereoSet**; 2021) created human annotated contexts of social bias types, while Nangia et al. (2020) proposed Crowdsourced Stereotype Pairs benchmark (**CrowS-**

Pairs). Following this prior work, we define a stereotype as a commonly-held association between a group and some attribute. These benchmarks use sentence pairs of the form "*She is a nurse/doctor*". StereoSet calculates log-odds by masking the modified tokens (*nurse*, *doctor*) in a sentence pair, whereas CrowS-Pairs calculates log-odds by masking their unmodified tokens (*She*, *is*, *a*).

Kaneko and Bollegala (2022a) proposed All Unmasked Likelihood (**AUL**) and AUL with Attention weights (**AULA**), which calculate log-likelihood by predicting all tokens in a test case, given the contextualised embedding of the unmasked input.

5.4 Evaluation Metrics for Social Biases in Static Sense Embeddings

We extend the WEAT and WAT datasets that have been frequently used in prior work for evaluating social biases in static word embeddings such that they can be used to evaluate sense embeddings. These datasets compare the association between a target word w and some (e.g. pleasant or unpleasant) attribute a, using the cosine similarity, $\cos(\boldsymbol{w}, \boldsymbol{a})$, computed using the static word embeddings $\boldsymbol{w}$ and $\boldsymbol{a}$ of respectively w and a. Given two same-sized sets of *target* words $\mathcal{X}$ and $\mathcal{Y}$ and two sets of *attribute* words $\mathcal{A}$ and $\mathcal{B}$, the bias score, $s(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B})$, for each target is calculated as follows:

$$s(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}) = \sum_{\boldsymbol{x} \in \mathcal{X}} w(\boldsymbol{x}, \mathcal{A}, \mathcal{B}) - \sum_{\boldsymbol{y} \in \mathcal{Y}} w(\boldsymbol{y}, \mathcal{A}, \mathcal{B}) \tag{5.1}$$

$$w(\boldsymbol{t}, \mathcal{A}, \mathcal{B}) = \operatorname*{mean}_{\boldsymbol{a} \in \mathcal{A}} \cos(\boldsymbol{t}, \boldsymbol{a}) - \operatorname*{mean}_{\boldsymbol{b} \in \mathcal{B}} \cos(\boldsymbol{t}, \boldsymbol{b}) \tag{5.2}$$

Here, $\cos(\boldsymbol{a}, \boldsymbol{b})$ is the cosine similarity[1] between the embeddings $\boldsymbol{a}$ and $\boldsymbol{b}$. The one-sided p-value for the permutation test for $\mathcal{X}$ and $\mathcal{Y}$ is calculated as the probability of $s(\mathcal{X}_i, \mathcal{Y}_i, \mathcal{A}, \mathcal{B}) > s(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B})$. The effect size is calculated as the normalised measure given by (5.3):

$$\frac{\underset{x \in \mathcal{X}}{\text{mean}}\, w(x, \mathcal{A}, \mathcal{B}) - \underset{y \in \mathcal{Y}}{\text{mean}}\, w(y, \mathcal{A}, \mathcal{B})}{\underset{t \in \mathcal{X} \cup \mathcal{Y}}{\text{sd}}\, w(t, \mathcal{A}, \mathcal{B})} \tag{5.3}$$

We repurpose these datasets for evaluating the social biases in *sense* embeddings as follows. For each target word in WEAT, we compare each sense s_i of the target word w against each sense a_j of a word selected from the association graph using their corresponding sense embeddings, $\boldsymbol{s}_i, \boldsymbol{a}_j$, and use the maximum similarity over all pairwise combinations (i.e. $\max_{i,j} \cos(\boldsymbol{s}_i, \boldsymbol{a}_j)$) as the word association measure. Measuring similarity between two words as the maximum similarity over all candidate senses of each word is based on the assumption that two words in a word-pair would mutually disambiguate each other in an association-based evaluation (Pilehvar and Camacho-Collados, 2019), and has been used as a heuristic for disambiguating word senses (Reisinger and Mooney, 2010).

WAT considers only gender bias and calculates the gender information vector for each word in a word association graph created with Small World of Words project (Deyne et al., 2019) by propagating information related to masculine and feminine words $(w_m^i, w_f^i) \in \mathcal{L}$ using a random walk approach (Zhou et al., 2003). It is non-trivial to pre-specify the sense of a word in a large word association graph considering the paths followed by a random walk. The gender information is encoded as a vector (b_m, b_f) in 2 dimensions, where b_m and b_f denote the masculine and feminine orientations of a word, respectively. The bias score of a word is defined as $\log(b_m/b_f)$. The gender bias of word embeddings

[1] Alternatively, inner-products can be used to extend RIPA.

Category	noun vs. verb	race vs. colour	nationality vs. language
#pleasant words	14	5	18
#unpleasant words	18	5	15
#target words	6	1	16
#templates	1	4	4
#test cases	324	733	2304

Table 5.1: Statistics of the the SSSB dataset.

is evaluated using the Pearson correlation coefficient between the bias score of each word and the score given by (5.4), computed as the average over the differences of cosine similarities between masculine and feminine words.

$$\frac{1}{|\mathcal{L}|}\sum_{i=1}^{|\mathcal{L}|}\left(\cos(\boldsymbol{w},\boldsymbol{w}_m^i)-\cos(\boldsymbol{w},\boldsymbol{w}_f^i)\right) \tag{5.4}$$

To evaluate gender bias in sense embeddings, we follow the method that is used in WEAT, and take $\max_{i,j}\cos(\boldsymbol{s}_i,\boldsymbol{a}_j))$ as the word association measure.

5.5 Sense-Sensitive Social Bias Dataset

Contextualised embeddings such as the ones generated by MLMs return different vectors for the same word in different contexts. However, the datasets discussed in § 5.4 do not provide contextual information for words and cannot be used to evaluate contextualised embeddings. Moreover, the context in which an ambiguous word occurs determines its word sense. Contextualised sense embedding methods such as SenseBERT (fine-tuned using WordNet super senses) have been shown to cap-

Category	Ambiguous words considered
noun vs. verb	engineer, carpenter, guide, mentor, judge, nurse
race vs. colour	black
nationality vs. language	Japanese, Chinese, English, Arabic, German, French, Spanish, Portuguese, Norwegian, Swedish, Polish, Romanian, Russian, Egyptian, Finnish, Vietnamese

Table 5.2: Bias categories covered in the SSSB dataset

ture word sense information in their contextualised embeddings (Zhou and Bollegala, 2021).

CrowS-Pairs and StereoSet datasets were proposed for evaluating contextualised word embeddings. Specifically, an MLM is considered to be unfairly biased if it assigns higher pseudo-log-likelihood scores for stereotypical sentences, S^{st}, than anti-stereotypical ones, S^{at}. However, both of those datasets do not consider multiple senses of words and cannot be used to evaluate social biases in contextualised sense embeddings.

To address this problem, we create the SSSB dataset, containing template-generated sentences covering multiple senses of ambiguous words for three types of social biases: *gender*, *race* and *nationality*. Templates are used in the same sense as in prior work such as Kurita et al. (2019). For example, we manually create templates such as [gender word] is a [pleasant/unpleasant attribute] engineer. We then fill the gender word by male and female gender pronouns (he/she), pleasant attributes (e.g. careful, skillful, efficient, etc.) and unpleasant attributes (e.g. clumsy, unskillful, inefficient, etc.) to generate many example sentences demonstrating social biases.

To the best of our knowledge, SSSB is the first-ever dataset created for the purpose of evaluating social biases in sense embeddings. Ta-

ble 5.1 shows the summary statistics of the SSSB dataset. Table 5.2 shows the bias categories covered in the SSSB dataset. Next, we describe the social biases covered in this dataset.

5.5.1 Nationality vs. Language Bias

These examples cover social biases related to nationality (racial) or language (non-racial). Each test case covers two distinct senses and the following example shows how they represent biases. *Japanese people are nice* is an anti-stereotype for *Japanese* as a nationality because it is associated with a pleasant attribute (i.e. *nice*) in this example sentence. On the other hand, *Japanese people are stupid* is a stereotype for *Japanese* as a nationality because it is associated with an unpleasant attribute (i.e. *stupid*). These can be considered as examples of racial biases.

Likewise, for the language sense of Japanese, we create examples as follows. *Japanese language is difficult to understand* is a stereotype for *Japanese* as a language because it is associated with an unpleasant attribute (i.e. *difficult*). On the other hand, *Japanese language is easy to understand* is an anti-stereotype for *Japanese* as a language because it is associated with a pleasant attribute (i.e. *easy*).

In SSSB, we indicate the sense-type, WordNet sense-id and the type of social bias in each example as follows:

> *Japanese people are beautiful.*
> [nationality, japanese%1:18:00::, anti]

Here, sense-type is nationality, sense-id, as specified in the WordNet, is *japanese%1:18:00::* and the bias is anti (we use the labels *anti* and *stereo* to denote respectively anti-stereotypical and stereotypical biases).

We use the likelihood scores returned by an MLM to nationality

vs. language sentence pairs as described further in § 5.6 to evaluate social biases in MLMs. Essentially, if the likelihood score returned by an MLM for the example that uses an unpleasant attribute is higher than the one that uses a pleasant attribute for a member in the disadvantaged group, then we consider the MLM to be socially biased. Moreover, if a member in the disadvantaged group is associated with a positive attribute in a stereotypical manner, we consider this as an anti-stereotype case. For example, we classify *Asians are smart* as anti-stereotype rather than "positive" stereotypes following prior work on word-level or sentence-level bias evaluation datasets (e.g., Crows-Pairs and StereoSet) to focus on more adverse types of biases that are more direct and result in discriminatory decisions against the disadvantaged groups.

Note that one could drop the modifiers such as *people* and *language* and simplify these examples such as *Japanese are nice* and *Japanese is difficult* to generate additional test cases. However, the sense-sensitive embedding methods might find it difficult to automatically disambiguate the correct senses without the modifiers such as *language* or *people*. Therefore, we always include these modifiers when creating examples for nationality vs. language bias in the SSSB dataset.

5.5.2 Race vs. Colour Bias

The word *black* can be used to represent the race (black people) or the colour. We create examples that distinguish these two senses of black as in the following example. *Black people are friendly* represents an anti-stereotype towards *black* because it is associated with a pleasant attribute (i.e. *friendly*) of a disadvantaged group whereas, *Black people are arrogant* represents a stereotype because it is associated with an unpleasant attribute (i.e. *arrogant*).

On the other hand, for the colour black, *The black dress is elegant*

represents an anti-stereotype because it is associated with a pleasant attribute (i.e. *elegant*), whereas *The black dress is ugly* represents a stereotype because it is associated with an unpleasant attribute (i.e. *ugly*). If the likelihood score returned by an MLM for a sentence containing the racial sense with an unpleasant attribute is higher than one that uses a pleasant attribute, the MLM is considered to be socially biased.

5.5.3 Gender Bias in Noun vs. Verb Senses

To create sense-related bias examples for gender,[2] we create examples based on occupations. In particular, we consider the six occupations: *engineer*, *nurse*, *judge*, *mentor*, *(tour) guide*, and *carpenter*. These words can be used in a noun sense (e.g. *engineer is a person who uses scientific knowledge to solve practical problems*, *nurse is a person who looks after patients*, etc.) as well as in a verb sense expressing the action performed by a person holding the occupation (e.g. *design something as an engineer*, *nurse a baby*, etc.). Note that the ambiguity here is in the occupation (noun) vs. action (verb) senses and not in the gender, whereas the bias is associated with the gender of the person holding the occupation.

To illustrate this point further, consider the following examples. *She is a talented engineer* is considered as an anti-stereotypical example for the noun sense of *engineer* because females (here considered as the disadvantaged group) are not usually associated with pleasant attributes (i.e. *talented*) with respect to this occupation (i.e. *engineer*). *He is a talented engineer* is considered as a stereotypical example for the noun sense of engineer because males (here considered as the advantaged group) are usually associated with pleasant attributes with regard to this occupation. As described in § 5.6, if an MLM assigns a higher

[2]We consider only male and female genders in this work

likelihood to the stereotypical example (second sentence) than the anti-stereotypical example (first sentence), then that MLM is considered to be gender biased.

On the other hand, *She is a clumsy engineer* is considered to be a stereotypical example for the noun sense of engineer because females (i.e. disadvantaged group) are historically associated with such unpleasant attributes (i.e. *clumsy*) with respect to such male-dominated occupations. Likewise, *He is a clumsy engineer* is considered as an anti-stereotypical example for the noun sense of engineer because males (i.e. advantaged group) are not usually associated with such unpleasant attributes (i.e. *clumsy*). Here again, if an MLM assigns a higher likelihood to the stereotypical example (first sentence) than the anti-stereotypical example (second sentence), then it is considered to be gender biased. Note that the evaluation direction with respect to male vs. female pronouns used in these examples is opposite to that in the previous paragraph because we are using an unpleasant attribute in the second set of examples.

Verb senses are also used in sentences that contain gender pronouns in SSSB. For example, for the verb sense of *engineer*, we create examples as follows: *She used novel material to engineer the bridge.* Here, the word engineer is used in the verb sense in a sentence where the subject is a female. The male version of this example is as follows: *He used novel material to engineer the bridge.* In this example, a perfectly unbiased MLM should not systematically prefer one sentence over the other between the two sentences both expressing the verb sense of the word *engineer*.

5.6 Evaluation Metrics for Social Biases in Contextualised Sense Embeddings

For a contextualised (word/sense) embedding under evaluation, we compare its pseudo-likelihood scores for stereotypical and anti-stereotypical sentences for each sense of a word in SSSB, using AUL (Kaneko and Bollegala, 2022a).[3] AUL is known to be robust against the frequency biases of words and provides more reliable estimates compared to the other metrics for evaluating social biases in MLMs. Following the standard evaluation protocol, we provide AUL the complete sentence $S = w_1, \ldots, w_{|S|}$, which contains a length $|S|$ sequence of tokens w_i, to an MLM with pretrained parameters θ. We first compute PLL(S), the Pseudo Log-Likelihood (PLL) for predicting all tokens in S excluding begin and end of sentence tokens, given by (5.5):

$$\text{PLL}(S) \coloneqq \frac{1}{|S|} \sum_{i=1}^{|S|} \log P(w_i|S;\theta) \tag{5.5}$$

Here, $P(w_i|S;\theta)$ is the probability assigned by the MLM to token w_i conditioned on S. The fraction of sentence-pairs in SSSB, where higher PLL scores are assigned to the stereotypical sentence than the anti-stereotypical one is considered as the AUL *bias score* of the MLM associated with the contextualised embedding, and is given by (5.6):

$$\text{AUL} = \left(\frac{100}{N} \sum_{(S^{\text{st}}, S^{\text{at}})} \mathbb{I}(\text{PLL}(S^{\text{st}}) > \text{PLL}(S^{\text{at}})) \right) - 50 \tag{5.6}$$

[3] The attention-weighted variant (AULA) is not used because contextualised sense embeddings have different structures of attention from contextualised embeddings, and it is not obvious which attention to use in the evaluations.

Here, N is the total number of sentence-pairs in SSSB and $\mathbb{I}$ is the indicator function, which returns 1 if its argument is True and 0 otherwise. AUL score given by (5.6) falls within the range $[-50, 50]$ and an unbiased embedding would return bias scores close to 0, whereas bias scores less than or greater than 0 indicate bias directions towards respectively the anti-stereotypical or stereotypical examples.

5.7 Experiments and Results

In this section, we evaluate social biases in both static and contextualised sense embeddings.

5.7.1 Bias in Static Embeddings

In order to evaluate biases in static sense embeddings, we select two current SoTA sense embeddings: LMMS[4] (Loureiro and Jorge, 2019a) and ARES[5] (Scarlini et al., 2020b). In addition to WEAT and WAT datasets described in § 5.4, we also use SSSB to evaluate static sense embeddings using the manually assigned sense ids for the target and attribute words, ignoring their co-occurring contexts. LMMS and ARES sense embeddings associate each sense of a lexeme with a sense key and a vector, which we use to compute cosine similarities as described in § 5.4. To compare the biases in a static sense embedding against a corresponding sense-insensitive static word embedding version, we compute a static word embedding $\boldsymbol{w}$, for an ambiguous word w by taking the average (**avg**) over the sense embeddings $\boldsymbol{s}_i$ for all of w's word senses as given in (5.7), where $M(w)$ is the total number of senses of w:

$$\boldsymbol{w} = \frac{\sum_i^{M(w)} \boldsymbol{s_i}}{M(w)} \tag{5.7}$$

[4] https://github.com/danlou/LMMS
[5] http://sensembert.org

This would simulate the situation where the resultant embeddings are word-specific but not sense-specific, while still being comparable to the original sense embeddings in the same vector space. As an alternative to (5.7), which weights all different senses of w equally, we can weight different senses by their frequency. However, such sense frequency statistics are not always available except for sense labelled corpora such as SemCor (Miller et al., 1993). Therefore, we use the unweighted average given by (5.7).

From Table 5.3 we see that in WEAT[6] in all categories considered, sense embeddings always report a higher bias compared to their corresponding sense-insensitive word embeddings. This shows that even if there are no biases at the word level, we can still observe social biases at the sense level in WEAT. However, in the WAT dataset, which covers only gender-related biases, we see word embeddings to have higher biases than sense embeddings. This indicates that in WAT gender bias is more likely to be observed in static word embeddings than in static sense embeddings.

In SSSB, word embeddings always report the same bias scores for the different senses of an ambiguous word because static word embeddings are neither sense nor context sensitive. As aforementioned, the word "black" is bias-neutral with respect to the colour sense, while it often has a social bias for the racial sense. Consequently, for *black* we see a higher bias score for its racial than colour sense in both LMMS and ARES sense embeddings.

In the bias scores reported for *nationality* vs. *language* senses, we find that *nationality* obtains higher biases at the word level, while *language* at the sense level in both LMMS and ARES. Unlike *black*, where the two senses (colour vs. race) are distinct, the two senses *nationality*

[6]Three bias types (European vs. African American, Male vs. Female, and Old vs. Young) had to be excluded because these biases are represented using personal names that are not covered by LMMS and ARES sense embeddings.

	LMMS	ARES
Dataset	word/sense	word/sense
WEAT		
Flowers vs Insects	1.63/**2.00**	1.58/**2.00**
Instruments vs Weapons	1.42/**2.00**	1.37/**1.99**
Math vs Art	1.52/**1.83**	0.98/**1.45**
Science vs Art	1.38/**1.66**	0.92/**1.44**
Physical vs. Mental condition	0.42/**0.64**	-0.12/**-0.77**
WAT	**0.53**/0.41	**0.46**/0.31
SSSB		
black (race)	**5.36**/4.64	5.40/**5.67**
black (colour)	**5.36**/1.64	**5.40**/4.83
nationality	**7.78**/7.01	**6.94**/5.75
language	7.78/**8.23**	6.94/**7.38**
noun	0.34/**0.39**	0.09/**0.16**
verb	**0.34**/0.26	**0.09**/0.06

Table 5.3: Bias in LMMS and ARES Static Sense Embeddings. In each row, between sense-insensitive word embeddings and sense embeddings, the larger deviation from 0 is shown in bold. All results on WEAT are statistically signiciant ($p < 0.05$) according to (5.3).

and *language* are much closer because in many cases (e.g. Japanese, Chinese, Spanish, French etc.) languages and nationalities are used interchangeably to refer to the same set of entities. Interestingly, the *language* sense is assigned a slightly higher bias score than the *nationality* sense in both LMMS and ARES sense embeddings. Moreover, we see that the difference between the bias scores for the two senses in *colour* vs. *race* (for black) as well as *nationality* vs. *language* is more in LMMS compared to that in ARES sense embeddings.

Between noun vs. verb senses of occupations, we see a higher gender bias for the noun sense than the verb sense in both LMMS and ARES sense embeddings. This agrees with the intuition that gender biases

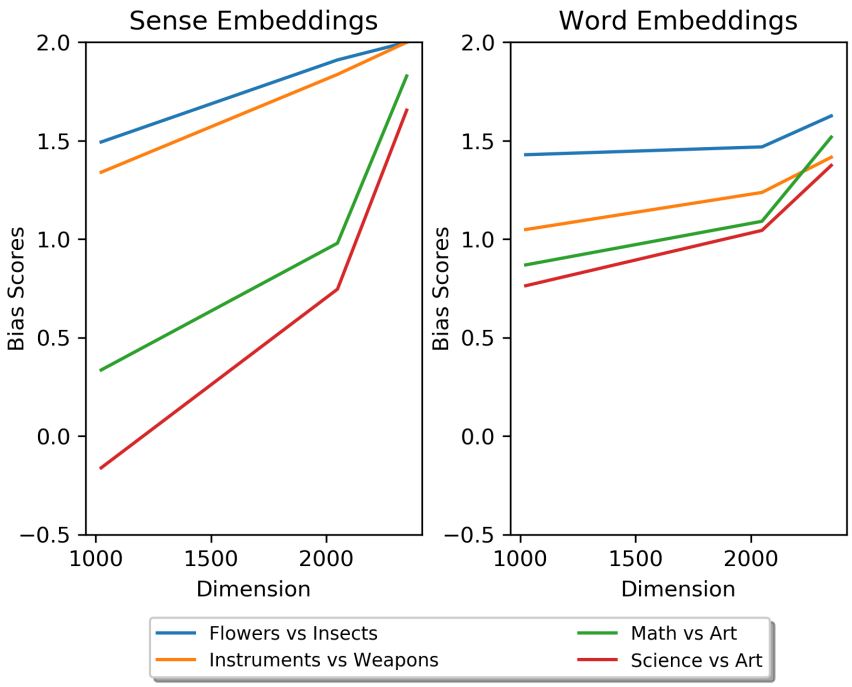


Figure 5.2: Effect of the dimensionality of sense embeddings (LMMS) and word embeddings (LMMS-average).

exist with respect to occupations and not so much regarding what actions/tasks are carried out by the persons holding those occupations. Compared to word embeddings, there is a higher bias for the sense embeddings in the noun sense for both LMMS and ARES. This trend is reversed for the verb sense where we see higher bias scores for the word embeddings than the corresponding sense embeddings in both LMMS and ARES. Considering that gender is associated with the noun than verb sense of occupations in English, this shows that there are hidden gender biases that are not visible at the word level but become more apparent at the sense level. This is an important factor to consider when evaluating gender biases in word embeddings, which has been largely ignored thus far in prior work.

To study the relationship between the dimensionality of the embed-

ding space and the social biases it encodes, we compare 1024, 2048 and 2348 dimensional LMMS static sense embeddings and their corresponding word embeddings (computed using (5.7)) on the WEAT dataset in Figure 5.2. We see that all types of social biases increase with the dimensionality for both word and sense embeddings. This is in agreement with Silva et al. (2021) who also reported that increasing model capacity in contextualised word embeddings does not necessarily remove their unfair social biases. Moreover, in higher dimensionalities sense embeddings show a higher degree of social biases than the corresponding (sense-insensitive) word embeddings.

5.7.2 Bias in Contextualised Embeddings

To evaluate biases in contextualised sense embeddings, we use SenseBERT[7], which is a fine-tuned version of BERT[8] to predict supersenses in the WordNet. For both BERT and SenseBERT, we use base and large pretrained models of dimensionalities respectively 768 and 1024. Using AUL, we compare biases in BERT and SenseBERT using SSSB, CrowS-Pairs and StereoSet[9] datasets. Note that unlike SSSB, CrowS-Pairs and StereoSet *do not* annotate for word senses, and hence cannot be used to evaluate sense-specific biases.

Table 5.4 compares the social biases in contextualised word/sense embeddings. For both base and large versions, we see that in CrowS-Pairs, BERT is more biased than SenseBERT, whereas the opposite is true in StereoSet. Among the nine bias types included in CrowS-Pairs, *gender* bias related test instances are the second most frequent following *racial* bias. On the other hand, gender bias related examples are relatively less frequent in StereoSet (cf. gender is the third most frequent bias type with 40 instances among the four bias types in StereoSet fol-

[7] https://github.com/AI21Labs/sense-bert
[8] https://github.com/huggingface/transformers
[9] We use only intrasentence test cases in StereoSet.

	base	large
Dataset	BERT/SenseBERT	BERT/SenseBERT
CrowS-Pairs	**-1.66**/0.99	**-3.58**/2.45
StereoSet	-1.09/**8.31**	-1.47/**6.51**
SSSB		
race	10.19/**14.81**	**-17.59**/0.00
colour	**-6.64**/-2.96	-8.88/**9.84**
nationality	5.79/**15.34**	4.28/**8.10**
language	-0.17/**-2.95**	**6.25**/-3.82
noun	10.42/**14.06**	**3.13**/**3.13**
verb	**12.89**/-3.74	0.22/**-15.44**

Table 5.4: Bias in BERT and SenseBERT contextualised word/sense embeddings. In each row, between the AUL bias scores for the word vs. sense embeddings, the larger deviation from 0 is shown in bold.

lowing *race* with 149 instances and *profession* with 120 instances out of the total 321 intrasentence instances). This difference in the composition of bias types explains why the bias score of BERT is higher in CrowS-Pairs, while the same is higher for SenseBERT in StereoSet.

In SSSB, in 8 out of the 12 cases, SenseBERT demonstrates equal or higher absolute bias scores than BERT. This result shows that even in situations where no biases are observed at the word level, there can still be significant degrees of biases at the sense level. In some cases (e.g. *verb* sense in base models and *colour*, *language* and *verb* senses for the large models), we see that the direction of the bias is opposite between BERT and SenseBERT. Moreover, comparing with the corresponding bias scores reported by the static word/sense embeddings in Table 5.3, we see higher bias scores reported by the contextualised word/sense embeddings in Table 5.4. Therefore, we recommend future work studying social biases to consider not only word embedding models but also sense embedding models.

stereo/anti-stereo sentences	BERT stereo	BERT anti	BERT diff	SenseBERT stereo	SenseBERT anti	SenseBERT diff
he/she is a strong nurse	-0.45	-0.67	0.22	-15.71	-16.64	0.93
he/she is a professional nurse	-0.73	-0.85	0.11	-16.53	16.81	0.27
As a mother/father of five, she/he carefully nurse all of her/his children	-0.16	-0.15	-0.01	-18.07	-18.24	0.18
she/he made milk herself/himself to nurse the crying baby	-0.77	-0.14	-0.63	-15.85	-17.80	1.96

Table 5.5: Pseudo log-likelihood scores computed using Eq. (5.5) for stereo and anti-stereo sentences (shown together due to space limitations) using BERT-base and SenseBERT-base models. Here, diff = stereo - anti.

5.8 Gender Biases in SSSB

In this section, we further study the gender-related biases in static and contextualised word and sense embeddings using the noun vs. verb sense instances (described in § 5.5.3) in the SSSB dataset. To evaluate the gender bias in contextualised word/sense embeddings we use AUL on test sentences in SSSB noun vs. verb category. To evaluate the gender bias in static embeddings, we follow Bolukbasi et al. (2016) and use the cosine similarity between (a) the static word/sense embedding of the occupation corresponding to its noun or verb sense and (b) the gender directional vector $\boldsymbol{g}$, given by (5.8):

$$\boldsymbol{g} = \frac{1}{|\mathcal{C}|} \sum_{(m,f)\in\mathcal{C}} (\boldsymbol{m} - \boldsymbol{f}) \tag{5.8}$$

Here, (m, f) are male-female word pairs used by Kaneko and Bollegala (2019) such as (*he*, *she*) and $\boldsymbol{m}$ and $\boldsymbol{f}$ respectively denote their word embeddings. Corresponding sense-insensitive word embeddings are computed for the 2048 dimensional LMMS embeddings using (5.7).

Figure 5.3 shows the gender biases in LMMS embeddings. Because static word embeddings are not sense-sensitive, they report the same bias scores for both noun and verb senses for each occupation. For all noun senses, we see positive (male) biases, except for *nurse*, which is strongly female-biased. Moreover, compared to the noun senses, the

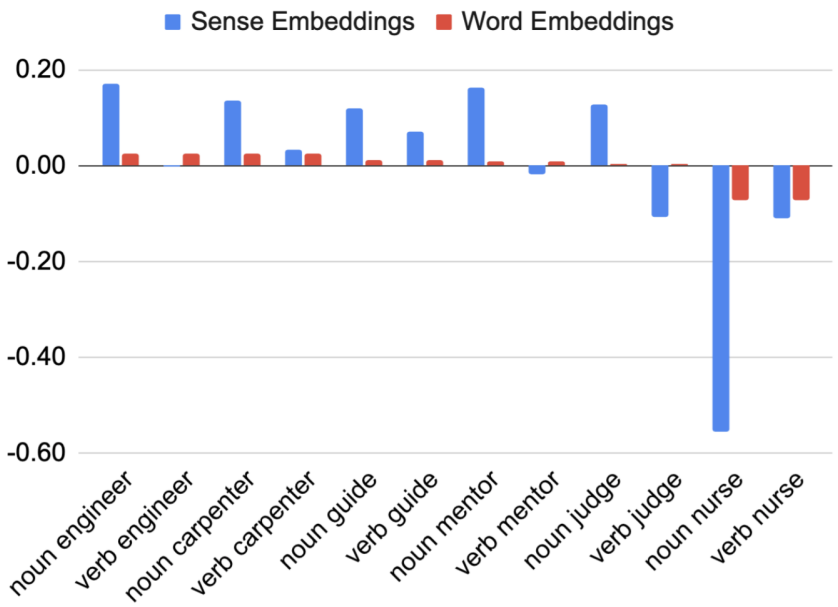


Figure 5.3: Gender biases found in the 2048-dimensional LMMS static sense embeddings and corresponding word embeddings computed using (5.7). Positive and negative cosine similarity scores with the gender directional vector (computed using (5.8)) represent biases towards respectively the *male* and *female* genders.

verb senses of LMMS are relatively less gender biased. This agrees with the intuition that occupations and not actions associated with those occupations are related to gender and hence can encode social biases. Overall, we see stronger biases in sense embeddings than in word embeddings.

Figure 5.4 shows the gender biases in BERT/SenseBERT embeddings. Here again, we see that for all noun senses, there are high stereotypical biases in both BERT and SenseBERT embeddings, except for *nurse* where BERT is slightly anti-stereotypically biased whereas SenseBERT shows a similar in magnitude but a stereotypical bias. Recall that *nurse* is stereotypically associated with the female gender, whereas other occupations are predominantly associated with males, which is reflected

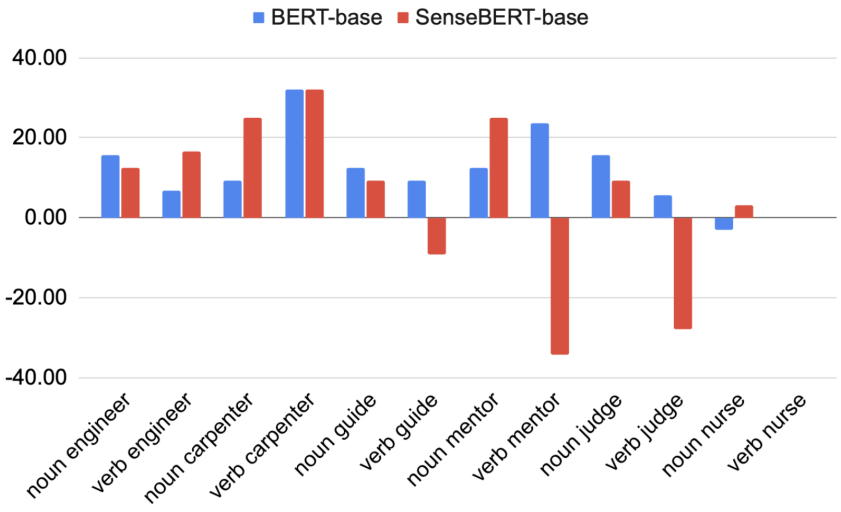


Figure 5.4: Gender biases found in 768-dimensional BERT-base and SenseBERT-base contextualised embeddings. Positive and negative AUL scores represent bias towards respectively the stereotypical and anti-stereotypical sentences.

in the AUL scores here.

Despite being not fine-tuned on word senses, BERT shows different bias scores for noun/verb senses, showing its ability to capture sense-related information via contexts. The verb sense embeddings of SenseBERT of *guide*, *mentor* and *judge* are anti-stereotypical, while the corresponding BERT embeddings are stereotypical. This shows that contextualised word and sense embeddings can differ in both magnitude as well as direction of the bias. Considering that SenseBERT is a fine-tuned version of BERT for a specific downstream NLP task (i.e. super-sense tagging), one must not blindly assume that an unbiased MLM to remain as such when fine-tuned on downstream tasks. *How social biases in word/sense embeddings change when used in downstream tasks* is an important research problem in its own right, which is beyond the scope of this thesis.

A qualitative analysis is given in Table 5.5 where the top-two sen-

tences selected from SSSB express the noun sense of *nurse*, whereas the bottom-two sentences express its verb sense. From Table 5.5, we see that SenseBERT has a higher preference (indicated by the high pseudo-log-likelihood scores) for stereotypical examples than BERT over anti-stereotypical ones (indicated by the higher diff values).

5.9 Summary

In this chapter, we evaluated social biases in sense embeddings by extending existing word-level bias evaluation datasets (WEAT, WAT) and by creating a novel sense-specific contextualised dataset (SSSB). Our experiments show that sense embeddings are also socially biased similar to word embeddings. Extending the analysis beyond English and developing debiasing methods for sense embedding are identified as important future research directions. In the next chapter, we will investigate the relationship between sense frequency and the ℓ_2 norm of sense embeddings.

6

ℓ_2 norm of sense embeddings

6.1 Introduction

Background: Given a text corpus, static word embedding learning methods (Pennington et al. 2014, Mikolov et al. 2013a, etc.) learn a *single* vector (aka *embedding*) to represent the meaning of a word in the corpus. In contrast, static sense embedding learning methods (Loureiro and Jorge 2019a, Scarlini et al. 2020b, etc.) learn *multiple* embeddings for each word, corresponding to the different senses of that word.

Arora et al. (2016) proposed a random walk model on the word co-occurrence graph and showed that if word embeddings are uniformly distributed over the unit sphere, the log-frequency of a word in a corpus is proportional to the squared ℓ_2 norm of the static word embedding, learned from the corpus. Hashimoto et al. (2016) showed that under a simple metric random walk over words where the probability of transitioning from one word to another depends only on the squared Euclidean distance between their embeddings, the log-frequency of word co-occurrences between two words converges to the negative squared Euclidean distance measured between the corresponding word embeddings. Mu and Viswanath (2018) later showed that word embeddings are distributed in a narrow cone, hence not satisfying the uniformity assump-

tion used by Arora et al. (2016), however their result still holds for such anisotropic embeddings. On the other hand, Arora et al. (2018) showed that word embedding can be represented as the linearly-weighted combination of sense embeddings. However, to the best of our knowledge, it remains unknown thus far as to **What is the relationship between the sense embeddings and the frequency of a sense?**, the central question that we study in this thesis.

Contributions: First, by extending the prior results for word embeddings into sense embeddings, we show that the **squared ℓ_2 norm of a static sense embedding is proportional to the log-frequency of the sense in the training corpus.** This finding has important practical implications. For example, it is known that assigning every occurrence of an ambiguous word in a corpus to the most frequent sense of that word (popularly known as the MFS baseline) is a surprisingly strong baseline for WSD (McCarthy et al., 2004, 2007). Therefore, the theoretical relationship which we prove implies that we should be able to use the ℓ_2 norm to predict the MFS of a word.

Second, we conduct a series of experiments to empirically validate the above-mentioned relationship. We find that the relationship holds for different types of static sense embeddings learned using methods such as GloVe and SGNS on SemCor.

Third, motivated by our finding that ℓ_2 norm of pretrained static sense embeddings encode sense-frequency related information, we use ℓ_2 norm of sense embeddings as a feature for several sense-related tasks such as (a) to predict the MFS of an ambiguous word, (b) determining whether the same sense of a word has been used in two different contexts (WiC), and (c) disambiguating the sense of a word in a sentence (WSD). We find that, regardless of its simplicity, ℓ_2 norm is a surprisingly effective feature, consistently improving the performance in all those benchmarks/tasks. The evaluation scripts are available at:

https://github.com/LivNLP/L2norm-of-sense-embeddings.

6.2 ℓ_2 NORM VS. FREQUENCY

Let us first revisit the generative model proposed by Arora et al. (2016) for static word embeddings, where the t-th word, v, in a corpus is generated at step t of a random walk of a *context* vector $\boldsymbol{c}_t$, which represents what is being talked about. The probability, $p(v|c_t)$, of emitting v at time t is modeled using a log-linear word production model, proportionally to $\exp({\boldsymbol{c}_t}^\top \boldsymbol{v})$. If $\mathcal{G}_v$ is a *word co-occurrence* graph, where vertices correspond to the words in the vocabulary, $\mathcal{V}$, the random walker can be seen as visiting the vertices in $\mathcal{G}_v$ according to this probability distribution. Arora et al. (2016) showed that the partition function, Z_c, given by (6.1) for this probabilistic model is a constant Z, independent of the context c.

$$Z_c = \sum_v \exp(\boldsymbol{c}^\top \boldsymbol{v}) \tag{6.1}$$

Assuming that the stationary distribution of this random walk is uniform over the unit sphere, Arora et al. (2016) proved the relationship in (6.2), for d dimensional word embeddings, $\boldsymbol{v} \in \mathbb{R}^d$.

$$\log p(v) = \frac{||\boldsymbol{v}||_2^2}{2d} - \log Z \tag{6.2}$$

Let the frequency of v in the corpus be $f(v)$, and the total number of word occurrences be $N = \sum_v f(v)$. $p(v)$ can be estimated using corpus counts as $f(v)/N$. Because N, d, and Z are constants, independent of v, (6.2) implies a linear relationship between $\log f(v)$ and $||\boldsymbol{v}||_2^2$.

To extend this result to sense embeddings, we observe that the word v generated at step t by the above-described random walk can be uniquely associated with a sense id s_v, corresponding to the meaning

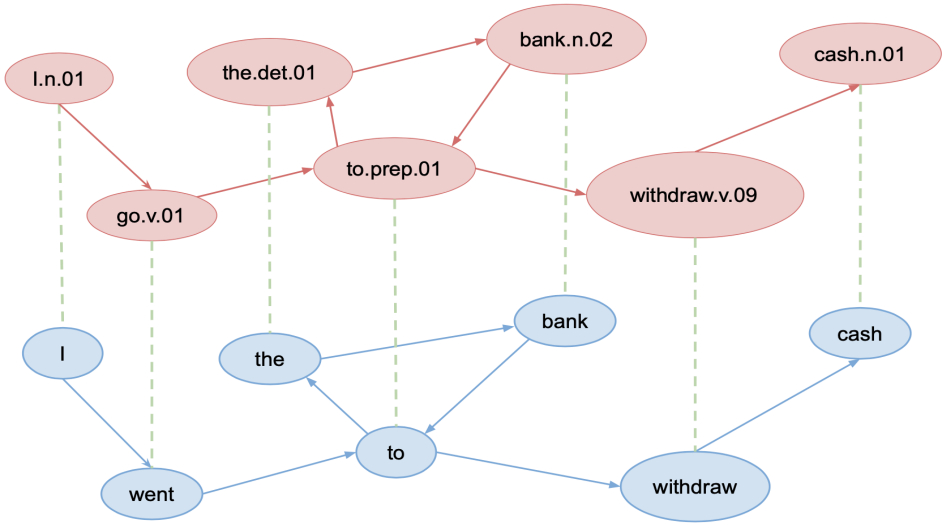


Figure 6.1: Part of the word co-occurrence graph $\mathcal{G}_v$ (bottom) shown with the corresponding sense co-occurrence graph $\mathcal{G}_s$ (top). Each word in $\mathcal{G}_v$ is mapped to its correct sense in $\mathcal{G}_s$.

of v as used in c_t. If we consider a second *sense co-occurrence* graph $\mathcal{G}_s$, where vertices correspond to the sense ids, then the above-mentioned corpus generation process corresponds to a second random walk on $\mathcal{G}_s$, as shown in Figure 6.1.

Although an ambiguous word can be mapped to multiple sense ids across the corpus in different contexts, at any given time step t, a word v is mapped to only one vertex in $\mathcal{G}_s$, determined by the context c_t. Indeed a WSD can be seen as the process of finding such a mapping. The two random walks over the word and sense id graphs are isomorphic and converge to the same set of final states (Bauerschmidt et al., 2021). Therefore, an analogous relationship given by (6.3) can be obtained by replacing word embeddings, $\boldsymbol{v}$, with sense embeddings, $\boldsymbol{s}$, in (6.2).

$$\log p(s) = \frac{||\boldsymbol{s}||_2^2}{2d_s} - \log Z' \tag{6.3}$$

Here, d_s is the dimensionality of the sense embeddings $\boldsymbol{s} \in \mathbb{R}^{d_s}$. Later in § 4.3, we empirically show that the normalisation coefficient, $Z' = \sum_s \exp(\boldsymbol{c}^\top \boldsymbol{s})$, for sense embeddings also satisfies the self-normalising (Andreas and Klein, 2015) property, thus independent of c. If we abuse the notation $f(s)$ to denote also the frequency of s in the corpus (i.e. the total number of times the random walker visits the vertex s), from (6.3) it follows that $\log f(s)$ is linearly related to $||\boldsymbol{s}||_2^2$.

6.3 Empirical Validation

The theoretical analysis described in § 6.2 implies a linear relationship between $\log f(s)$ and $||\boldsymbol{s}||_2^2$ for the learned sense embeddings. To empirically verify this relationship, we learn static sense embeddings using GloVe and SGNS from SemCor, which is the largest corpus manually annotated with WordNet sense ids. Specifically, we consider the co-occurrences of senses instead of words for this purpose. To distinguish the sense embeddings learned from GloVe and SGNS from their word embeddings, we denote these by respectively **GloVe-sense** and **SGNS-sense**.

6.3.1 Training GloVe-sense and SGNS-sense

We train our GloVe-sense and SGNS-sense on SemCor training data. Specifically, for each target word w in a context c, we train a vector and assign the annotated sense label to it. For GloVe-sense, we use its Python-based implementation.[1] We set the co-occurrence window to 10 tokens, number of dimensions to 300 and the initial learning rate to 0.05 for the vanilla stochastic gradient descent. We train the embeddings for 30 epochs with 2 parallel threads. To train SGNS-sense, we use the

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

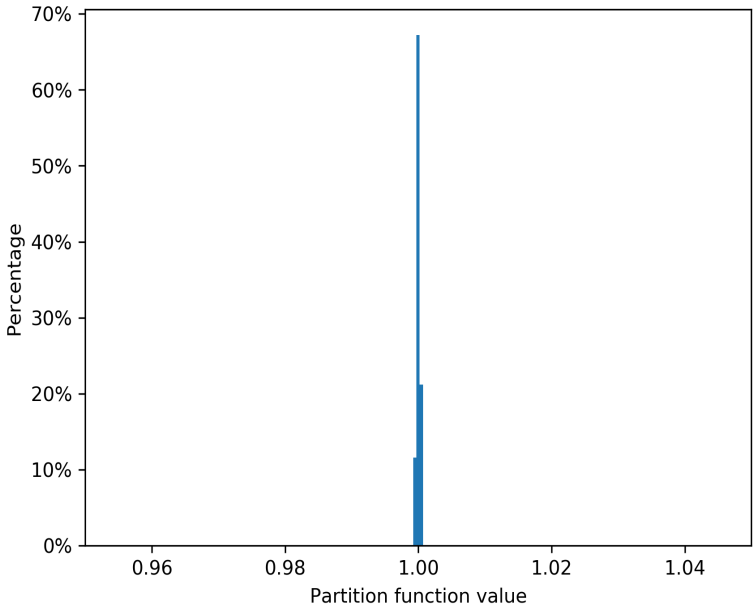


Figure 6.2: Histogram of the partition function for 1,000 random vectors $\boldsymbol{c}$ for GloVe-sense. The x-axis is normalised by the mean of the values.

Word2Vec module from gensim.models.[2] We set the min_count to 1 and the dimensionality of the embeddings to 300, and the remainder of the hyperparameters remain at their default values.

Figure 6.2 shows the partition function for GloVe-sense embeddings. We see that the partition function is tightly concentrated around its mean, showing that sense embeddings also demonstrate self-normalisation similar to word embeddings. Similar to the histogram for GloVe-sense embeddings, we see that the partition function for SGNS-sense embeddings is also tightly centred around the mean (i.e., 1.0) from Figure 6.3.

Figure 6.4 shows the correlation between $\log f(s)$ and $||\boldsymbol{s}||_2^2$ for GloVe-sense. We see a moderate positive correlation (Pearson's $\rho = 0.437$) between these two variables, confirming the linear relationship predicted in § 6.2. Similar to the correlation plot for GloVe-sense embeddings, from

[2] https://radimrehurek.com/gensim/models/word2vec.html

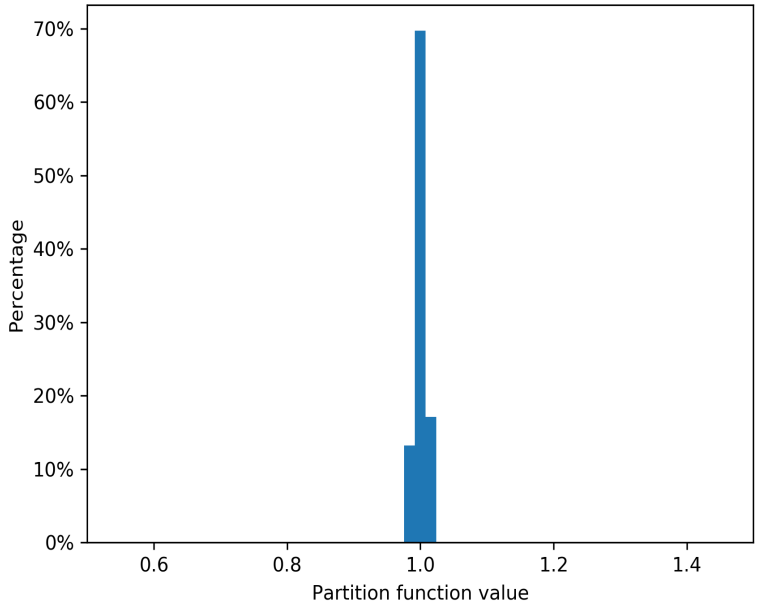


Figure 6.3: Histogram of partition function for 1,000 random vectors $\boldsymbol{c}$ for SGNS-sense. The x-axis is normalised by the mean of the values.

Figure 6.5, one can see a positive correlation (Pearson's $\rho = 0.440$) between the log-frequency and squared ℓ_2 norm for the SGNS-sense embeddings.

It is noteworthy however that this linear relationship between log-frequency and squared ℓ_2 norm does not hold for contextualised word embeddings such as BERT or static sense embeddings such as LMMS that are computed by averaging BERT embeddings (see § 6.4 for details). The random walk model described in § 6.2 cannot be applied to contextualised embeddings because the probability of occurrence of a word under the discriminative masked language modeling objectives used to train contextualised word embeddings such as BERT depends on all the words generated before as well as after the target word.

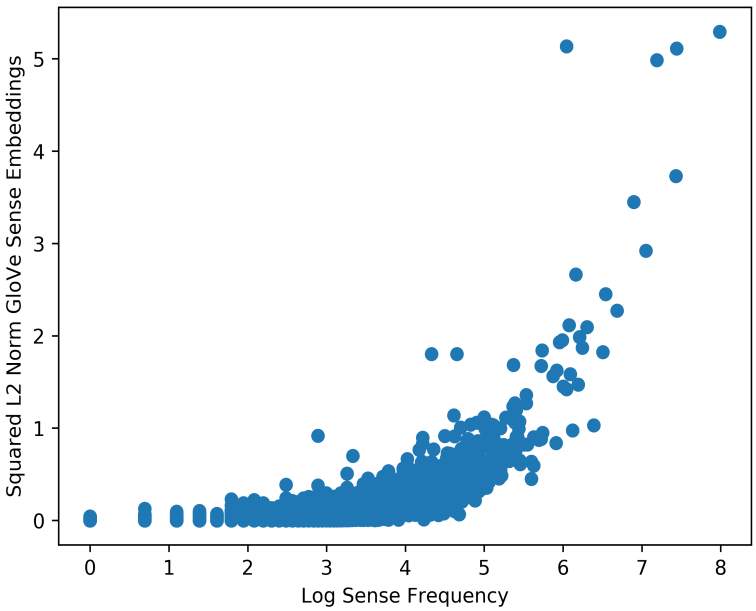


Figure 6.4: A linear relationship between $\log f(s)$ (x-axis) and $||\boldsymbol{s}||_2^2$ (y-axis) can be seen for GloVe-sense embeddings represented by the blue dots.

6.3.2 Predicting the Most Frequent Sense

To investigate whether the frequency of a sense is indeed represented by the squared ℓ_2 norm of its static sense embedding, we conduct an MFS prediction task on SemCor following the setup proposed by Hauer et al. (2019). In this MFS prediction task, given the set of senses of an ambiguous word, we must predict the sense with the highest frequency for that word in SemCor. For this purpose, we filter senses by the lemma and PoS of the target word and select the sense with the largest squared ℓ_2 norm using GloVe-sense and SGNS-sense embeddings separately.

Specifically, given a target word w in a context c, we first select a set of candidate senses based on w's lemma and PoS. Then we compute the ℓ_2 norm of the static sense embedding for each sense in the candidate set. Finally, we take the sense with the maximum ℓ_2 norm score as

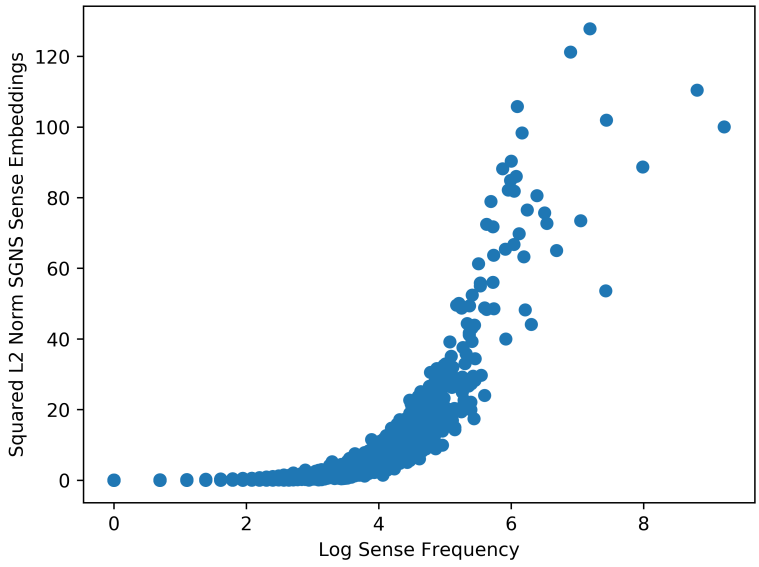


Figure 6.5: A linear relationship between $\log f(s)$ (x-axis) and $||\boldsymbol{s}||_2^2$ (y-axis) can be seen for SGNS-sense embeddings represented by the blue dots.

the predicted MFS for w. Then we compare our prediction with the MFS of w according to the sense occurrence in SemCor and compute the accuracy scores.

In Table 6.1, we compare our results against a random sense selection baseline and several prior proposals on the MFS benchmark dataset (Hauer et al., 2019). EnDi (Pasini and Navigli, 2018) is a language-independent and fully automatic method for sense distribution learning from raw text. UMFS-WE (Bhingardive et al., 2015) and WCT-VEC (Hauer et al., 2019) both use the distance between word and sense embeddings. COMP2SENSE (Hauer et al., 2019) is a knowledge-based method using WordNet and uses a set of words known as the companions of a target word to determine MFS, based on a sense-similarity function. As seen from Table 6.1, both GloVe-sense and SGNS-sense outperform all the other methods for **all** words and **noun** sample settings. In particular, for **noun** sample, which contains

Models	All words	Noun Sample
Random	67.6	26.0
UMFS-WE	73.9	48.0
EnDi	71.4	47.4
WCT-VEC	75.2	48.8
COMP2SENSE	77.9	58.5
Ours		
GloVe-sense with ℓ_2 norm	90.1	92.2
SGNS-sense with ℓ_2 norm	**95.6**	**96.6**

Table 6.1: Percentage accuracy for the MFS prediction task on SemCor for All Words and the Noun Sample, limited to polysemous nouns. Overall best scores are in bold.

polysemous nouns that occur at least 3 times in SemCor, both methods obtain more than 35% accuracy improvements over the next best method, providing strong empirical evidence supporting the linear relationship predicted by (6.3).

If the ℓ_2 norm of a sense embedding relates to the frequency of that sense, the ℓ_2 norm of the most frequent sense should be always greater than the ℓ_2 norm of the next frequent sense of an ambiguous word. To further investigate this, we sort the set of ambiguous words in SemCor based on their frequency and divide them into 10 subsets (i.e., bins). The summary statistics of each subset is shown in Table 6.2. For each ambiguous word w, we find its most frequent sense w_m and the next frequent sense w_n in SemCor. Note that w_m and w_n are determined based on their frequency in SemCor and not according to how they are sorted in WordNet.

Let us denote the ℓ_2 norms of w_m and w_n by $||\boldsymbol{w_m}||_2$ and $||\boldsymbol{w_n}||_2$ respectively, and $\alpha = \sum_{w \in \mathcal{V}} \mathbb{I}(||\boldsymbol{w_m}||_2 > ||\boldsymbol{w_n}||_2)$, where $\mathbb{I}(x)$ is the indicator function that returns 1 if x is True and 0 otherwise. $\mathcal{V}$ is the set of ambiguous words (i.e., the words that have a least two distinct senses in SemCor). We compute the percentage $\alpha/|\mathcal{V}|$ for the total vo-

Bins	Max Freq	Min Freq	Word Count
1	15,783	64	545
2	64	32	545
3	32	20	545
4	20	13	545
5	13	9	545
6	9	7	545
7	7	5	545
8	5	3	545
9	3	2	545
10	2	1	545

Table 6.2: Statistics of each bin of ambiguous words grouped based on their frequency in SemCor.

cabulary and show the result in Figure 6.6. We observe that the second bin obtains the highest $\alpha/|\mathcal{V}|$ score over all the bins. Moreover, the $\alpha/|\mathcal{V}|$ scores decrease with the frequency of the ambiguous words. This result indicates that the relationship between the ℓ_2 norm of a sense embedding and its frequency is stronger for high frequent words than low frequent ones.

6.3.3 Predicting Word Sense in Context

We evaluate the ℓ_2 norm of sense embeddings in WiC and WSD as downstream tasks. In WiC, given an ambiguous word w occurring in two contexts c_1 and c_2, we must predict if w occurs in c_1 and c_2 with the same sense or not. We follow Loureiro and Jorge (2019b), and train a binary logistic regression model on WiC training set using different sets of similarities between static sense embeddings and contextualised embeddings obtained from a language model (i.e. BERT) as features. We consider two current state-of-the-art sense embeddings, LMMS and ARES (Scarlini et al., 2020b), and include ℓ_2 norm of static sense em-

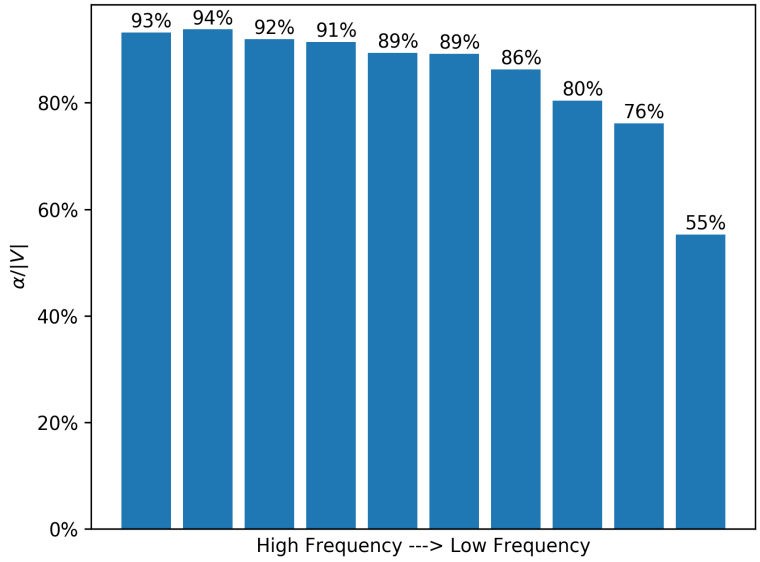


Figure 6.6: The trend of $\alpha/|\mathcal{V}|$ from high frequent words to low frequent words.

beddings as extra features, and measure the gain in performance.

We train a binary logistic regression classifier[3] on the WiC training set. To have a fair comparison against the original LMMS embeddings on WiC, we follow their work Loureiro and Jorge (2019b) to compute four similarities between sense and contextualised embeddings, and consider those as features. Specifically, given a target word w in two contexts c_1 and c_2, we first determine the sense-specific embeddings for w in c_1 and c_2, denoted by $\boldsymbol{s}_1(w)$ and $\boldsymbol{s}_2(w)$, as described in § 3.3.3. Then we use the cosine similarities between the two vectors in the following four pairs as features, requiring no expensive fine-tuning procedure:

1. $(\boldsymbol{s}_1(w), \boldsymbol{s}_2(w))$: the sense embedding of w in context c_1, $\boldsymbol{s}_1(w)$, and the sense embedding of w in context c_2, $\boldsymbol{s}_2(w)$.

2. $(\boldsymbol{t}(w, c_1), \boldsymbol{t}(w, c_2))$: the contextualised embedding of w in context

[3]We use the default parameters in scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

Models	Test
LMMS-based	
LMMS (Loureiro and Jorge, 2019a)	64.8
LMMS + ℓ_2 norm of GloVe-sense	65.8
LMMS+ℓ_2 norm of SGNS-sense	**67.0**
ARES-based	
ARES (Scarlini et al., 2020b)	66.6
ARES+ℓ_2 GloVe-sense	66.6
ARES+ℓ_2 SGNS-sense	66.7

Table 6.3: Accuracies on the WiC test sets for LMMS- (top) and ARES- (bottom) based classifiers. The overall best score is in bold.

c_1, $\boldsymbol{t}(w, c_1)$, and the contextualised embedding of w in context c_2, $\boldsymbol{t}(w, c_2)$.

3. $(\boldsymbol{s}_1(w), \boldsymbol{t}(w, c_1))$: the sense embedding of w in context c_1, $\boldsymbol{s}_1(w)$, and the contextualised embedding of w in context c_1, $\boldsymbol{t}(w, c_1)$.

4. $(\boldsymbol{s}_2(w), \boldsymbol{t}(w, c_2))$: the sense embedding of w in context c_2, $\boldsymbol{s}_2(w)$, and the contexualised embedding of w in context c_2, $\boldsymbol{t}(w, c_2)$.

Contextualised embeddings are not ℓ_2 normalised in this experiment. Here again, similar to the WSD settings described above, with respect to our proposed method, we simply append the ℓ_2 norm of the static sense embedding of w as the fifth feature.

From Table 6.3 we see that by including ℓ_2 norm of GloVe-sense and SGNS-sense embeddings as features, we obtain more than 1% gains in accuracy over the original LMMS on WiC test sets. ARES+ℓ_2 norm GloVe-sense obtains the same score as the ARES baseline, while ARES+ℓ_2 norm SGNS-sense achieves a slight improvement on the test set. This result shows that ℓ_2 norm of static sense embeddings encodes sense frequency related information, which improves the performance in WiC when used with static sense embeddings. This is noteworthy

given that ℓ_2 norm is a single feature compared to LMMS and ARES, which are both 2048 dimensional.

6.3.4 Word Sense Disambiguation

We further evaluate ℓ_2 norm of static sense embeddings using the English all-words WSD framework (Raganato et al., 2017). For this purpose, we train a binary logistic regression classifier using the two features – (a) the similarity between the contextualised embedding and a sense embedding of the target word, and (b) the squared ℓ_2 norm of the sense embedding. We use SemCor training data and consider the correct sense of the ambiguous target word as a positive instance, and its other senses as negative instances. At inference time, we predict the sense with the highest probability of being positive as the correct sense of the test word in the given context.

Specifically, we consider the Word Sense Disambiguation task as a binary classification problem and train a Logistic Regression binary classifier on SemCor. To evaluate the baselines, i.e., LMMS (LMMS SP-WSD: sensekeys[4]) and ARES on WSD, given a word w in a sentence c, we first compute its contextualised embedding using BERT (bert-large-cased) model by averaging the last four layers, denoted by $\boldsymbol{t}(w, c)$. We then compute the cosine similarity between $\boldsymbol{t}(w, c)$ and the sense embedding $\boldsymbol{s}(w)$ corresponding to the senses of w based on WordNet as a feature. We use the binary logistic regression classifier implemented in sklearn with the default parameters. For our proposed method, we simply append the ℓ_2 norm of static sense embedding of w as an additional feature. To avoid any discrepancies in the scoring methodology, we use the official scoring scripts of the English all-words WSD framework.

Likewise in the WiC evaluation in § 6.3.3, we measure the improvements in performance over LMMS and ARES, using ℓ_2 norm as a feature

[4] https://github.com/danlou/LMMS

Methods	SE2	SE3	SE07	SE13	S15	ALL
LMMS-based						
LMMS	76.3	75.6	68.1	75.1	77.0	75.4
LMMS+ℓ_2 norm GloVe-sense	77.8	76.9	70.5	76.6	77.8	76.8
LMMS+ℓ_2 norm SGNS-sense	77.5	77.4	69.7	77.1	78.1	76.9
ARES-based						
ARES	78.0	77.1	71.0	77.3	**83.2**	77.9
ARES+ℓ_2 norm GloVe-sense	**78.4**	**77.8**	**71.6**	77.9	82.4	**78.3**
ARES+ℓ_2 norm SGNS-sense	77.6	77.5	68.6	**78.0**	82.0	77.7

Table 6.4: F1 on the test sets of the all-words English WSD framework for LMMS- (top) and ARES- (bottom) based method. Overall best scores are in bold.

for WSD.

Table 6.4 shows the F1 scores for all-words English WSD datasets. ARES+ℓ_2 norm Glove-sense reports the best performance in three out of the five datasets and obtains the best performance on ALL (i.e., concatenation of all the test sets), whereas ARES+ℓ_2 norm SGNS-sense reports the best performance in SE13. In LMMS-based evaluations, we see that always either one or both GloVE/SGNS-sense ℓ_2 norms improve over the vanilla LMMS. This shows that we are able to improve the performance of both LMMS and ARES by simply adding the ℓ_2 norm of static sense embeddings as extra features.

6.4 Static Sense Embeddings from Contextualised Word Embeddings

We investigate whether the self-normalising and linearity properties hold for contextualised embeddings obtained from language models. For this purpose, we compute the static word embeddings for the words appearing in SemCor using contextualised embeddings learned by BERT.

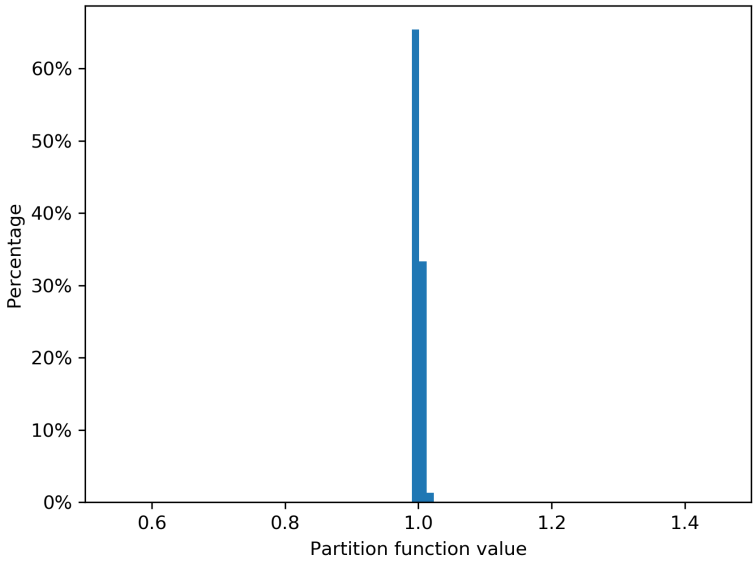


Figure 6.7: Histogram of the partition function for 1,000 random vectors c for BERT-static. The x-axis is normalised by the mean of the values.

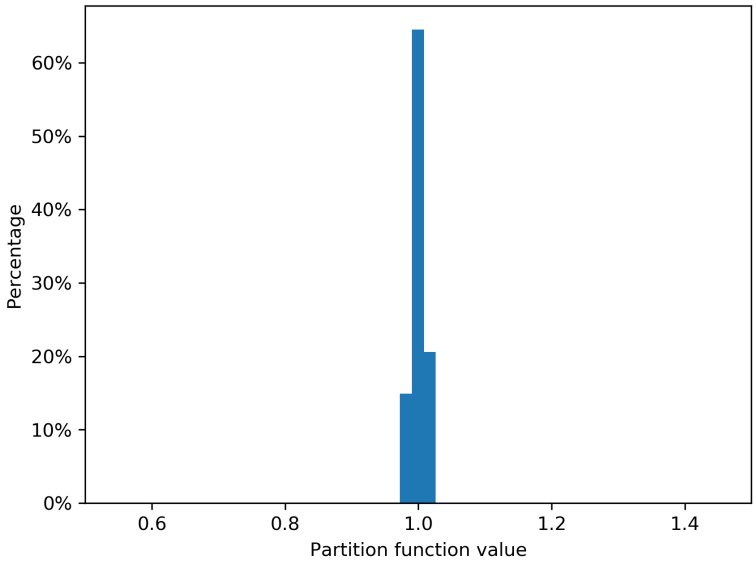


Figure 6.8: Histogram of the partition function for 1,000 random vectors c for LMMS. The x-axis is normalised by the mean of the values.

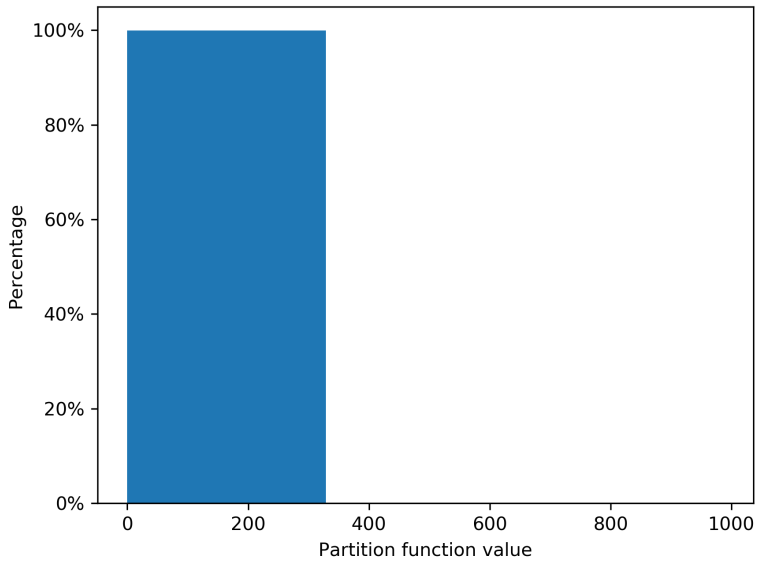


Figure 6.9: Histogram of the partition function for 1,000 random vectors c for LMMS_{sc}. The x-axis is normalised by the mean of the values.

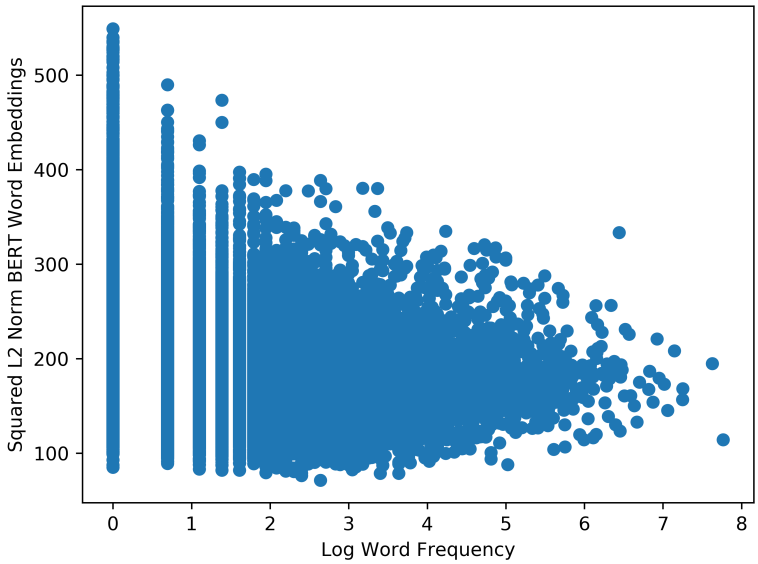


Figure 6.10: A linear relationship between $\log f(s)$ (x-axis) and $||\boldsymbol{s}||_2^2$ (y-axis) can be seen for BERT-static embeddings represented by the blue dots. The Pearson correlation coefficient between the two is -0.316.

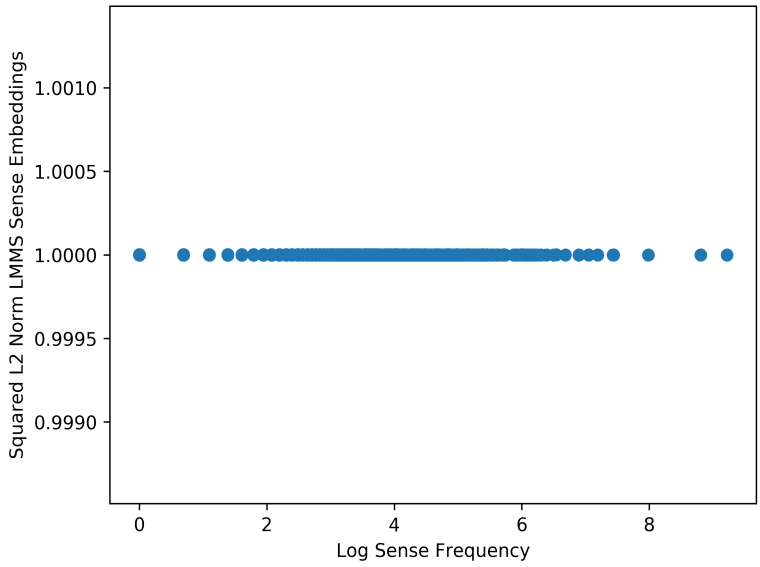


Figure 6.11: A linear relationship between $\log f(s)$ (x-axis) and $||\boldsymbol{s}||_2^2$ (y-axis) can be seen for LMMS embeddings represented by the blue dots. The Pearson correlation coefficient between the two is -0.005.

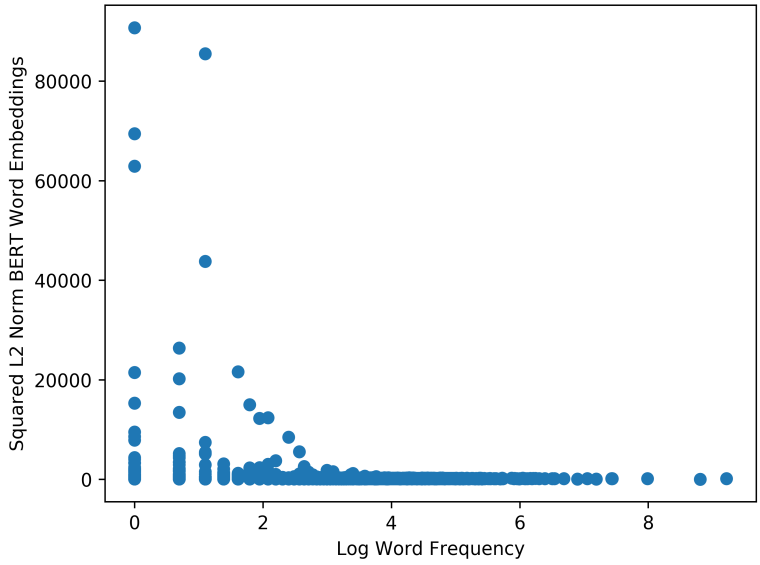


Figure 6.12: A linear relationship between $\log f(s)$ (x-axis) and $||\boldsymbol{s}||_2^2$ (y-axis) can be seen for LMMS$_{sc}$ embeddings represented by the blue dots. The Pearson correlation coefficient between the two is -0.010.

Specifically, we compute the average over the contextualised BERT embeddings for all of the occurrences of a word in SemCor and consider it as the static (i.e. context-independent) BERT embedding for that word. To distinguish the contextualised embeddings learned from BERT, we name the static BERT embeddings as BERT-static in the remainder of this thesis.

Recall that LMMS uses BERT to compute sense embeddings from SemCor and WordNet's glosses. Therefore, if BERT-static satisfies the self-normalising and linearity properties described in § 6.2, LMMS embedding must satisfy these properties as well. In addition, we take the first step of LMMS training procedure from the work of Loureiro and Jorge (2019a)[5] and train static sense embeddings only on SemCor data without normalising the learned sense embeddings (doing so would remove ℓ_2 norm related information from the sense embeddings). To differentiate this version of LMMS embeddings from the full-coverage LMMS embeddings, we refer to it as LMMS$_{sc}$ (here, sc stands for SemCor). We then test if the self-normalising and linearity properties hold for BERT-static, LMMS and LMMS$_{sc}$.

Figure 6.7, 6.8 and 6.9 show the histogram of partition functions for BERT-static, LMMS and LMMS$_{sc}$, respectively. We observe that the histograms of both BERT-static and LMMS are centred around the mean, while LMMS$_{sc}$ is not. This shows that LMMS$_{sc}$ does not satisfy self normalising, while BERT-static and LMMS do.

Figure 6.10, 6.11 and 6.12 show the correlation between squared ℓ_2 norms of the word/sense embeddings and the logarithms of word/sense frequencies for BERT-static, LMMS and LMMS$_{sc}$, respectively. From the figures, we see that none shows a linear relationship. This indicates that sense frequency related information is not encoded in the ℓ_2 norm of LMMS (or BERT) embeddings.

[5] https://github.com/danlou/LMMS

6.5 Summary

In this chapter, we investigated the relationship between the frequency and the ℓ_2 norm of a sense embedding and showed that the squared ℓ_2 norm of a static sense embedding is linearly related to its log frequency in the training corpus. Our experimental results indicate that, despite its simplicity, the ℓ_2 norm of sense embedding is a surprisingly effective feature for MFS prediction, WiC and WSD tasks. We made both theoretical and empirical contributions related to sense embeddings.

7 Conclusions and Future Work

This thesis was dedicated to the task of learning word sense representations and evaluating the properties of sense embeddings. This chapter concludes the work done in this thesis in the following sections. § 7.1 summarises the work presented in each chapter of the thesis. Next, an overview of the main findings and contributions of the thesis with respect to the research question and issues are reported in § 7.2. Finally, in § 7.3, some potential future directions that build upon the work conducted in the thesis are discussed.

7.1 Summary of Thesis

This thesis proposed multiple solutions to learning word sense embeddings, investigated different properties of sense embeddings and learn dynamic embeddings. Below we provide a concise summary of each chapter.

- Chapter 3 - **Learning Sense Embeddings**. In this chapter, we described our proposed Context Derived Embeddings of Senses (CDES), a method which is able to generate sense embeddings by extracting the sense-related information from contextualised

embeddings. To address the sense-coverage issue, CDES combines the gloss information from a semantic network with the data from an external corpus. Evaluations on several benchmark datasets for the WSD and WiC tasks demonstrate that CDES learns precise sense embeddings and produces results that are comparable to those of the present SoTA.

- Chapter 4 - **Learning Meta Sense Embeddings**. This chapter described the problem that not all existing sense embeddings cover all senses of ambiguous words equally well due to the discrepancies in their training resources. To address this problem, we propose the first-ever meta-sense embedding method – Neighbour Preserving Meta-Sense Embeddings, which learns meta-sense embeddings by combining multiple independently trained source sense embeddings such that the sense neighbourhoods computed from the source embeddings are preserved in the meta-embedding space. Our proposed method can combine source sense embeddings that cover different sets of word senses.

- Chapter 5 - **Social Biases in Sense Embeddings**. In this chapter, we considered the relatively underexplored aspect of social biases in pretrained sense embeddings. The biases in sense embeddings have received less attention than the many earlier studies evaluating the social biases in pretrained word embeddings. We created a new dataset for this purpose, which we name the Sense-Sensitive Social Bias (SSSB) dataset. The dataset we created is of a sensitive nature. We have included various sentences that express stereotypical biases associated with different senses of words in this dataset. We specifically considered three types of social biases in SSSB: (a) racial biases associated with nationality as opposed to language (e.g. Chinese people are cunning, Chinese language is difficult, etc.), (b) racial biases associated with the

word black as opposed to its sense as a colour (e.g. Black people are arrogant, Black dress is beautiful, etc.) and (c) gender-related biases associated with occupations used as nouns as opposed to verbs (e.g. She was a careless nurse, He was not able to nurse the crying baby, etc.).

- Chapter 6 - ℓ_2 **Norm of Sense Embeddings**. In this chapter, we inspected the relationship between the ℓ_2 norm of static sense embedding and its frequency in the training corpus. We evaluated the effectiveness of ℓ_2 norm of static sense embeddings on several experiments, i.e., MFS prediction, WiC and WSD tasks. We showed that the ℓ_2 norm of a static sense embedding encodes information related to the frequency of that sense in the training corpus used to learn the sense embeddings.

7.2 Contributions and Findings

In this section, we provide a synopsis of the main contributions and findings of the work in this thesis. To contextualise the contributions and findings, we re-emphasise the research questions associated with each chapter and then conclude with important findings. We began with Chapter 3.

1. *Can we extract sense-related information from contextualised word embeddings to create sense-specific versions of (pretrained) sense-agnostic static embeddings?*

   In Chapter 3, we found that contextualised embeddings produced by NLMs encode much more information beyond sense related information. Moreover, contextualised embeddings encode word sense related information that can be extracted and injected into sense-insensitive static word embeddings via (non)linear projections to create sense-sensitive versions of the sense-insensitive static

embeddings. We showed that our proposed CDES can accurately learn sense-specific static embeddings reporting comparable performance to the current SoTA sense embeddings.

2. *Can we learn sense embeddings that cover all senses of ambiguous words equally well, such that the sense-related information captured by the source sense embeddings can be preserved in the meta-sense embedding?*

   In Chapter 4, we answer this question by introducing NPMS. NPMS is able to learn meta-sense embeddings by combining multiple independently trained source sense embeddings such that the sense neighbourhoods computed from the source embeddings are preserved in the meta-embedding space. We showed that NPMS is an effective technique to increase the coverage of a pretrained sense embedding. In addition, NPMS does not require any projection matrix learning step and is able to create meta-embeddings even with sources of different dimensionalities.

3. *Can we create a method and a dataset to evaluate social biases at the sense level, even if there might be no biases found at the word level?*

   In Chapter 5, we showed that even if there are no biases at the word level, we can still observe social biases at the sense-level. To evaluate social biases in sense embeddings, we proposed SSSB dataset for evaluating social biases in sense embeddings. We found that contextualised word and sense embeddings can differ in both magnitude as well as direction of the bias. Moreover, we discovered that all types of social biases increase with the dimensionality for both word and sense embeddings.

4. *What is the relationship between the sense embeddings and the frequency of a sense?*

In Chapter 6, we found that the squared ℓ_2 norm of a static sense embedding is proportional to the log frequency of the sense in the training corpus. We showed that despite its simplicity, ℓ_2 norm of sense embedding is a surprisingly effective feature for MFS prediction, WiC and WSD tasks. Moreover, the relationship between the ℓ_2 norm of a sense embedding and its frequency is stronger for high frequent words than low frequent ones.

7.3 Future Work

Over the years, distributed semantic representations have shown to be effective to preserve prior information that may be integrated into downstream applications. This thesis focused on the representation of word senses, which can deal with the *meaning conflation deficiency* issue arising from representing a word with all its possible meanings as a single vector. This section outlines future research directions to the work described in this thesis.

7.3.1 Multilingual Approaches

All experiments and proposed methods described in this thesis are limited to the English language, which is morphologically limited. Therefore, the findings reported in this thesis might not generalise to other languages. On the other hand, there are already numerous multilingual MLMs such as mBERT (Devlin et al., 2019), XLM (CONNEAU and Lample, 2019) and XLM-R (Conneau et al., 2020), to name a few. Moreover, there are WSD and WiC benchmarks for other languages such as SemEval-13, SemEval-15, XL-WSD (Pasini et al., 2021) and WiC-XL (Raganato et al., 2020), as well as multilingual sense embeddings such as ARES$_m$ (Scarlini et al., 2020b) and SensEmBERT (Scarlini et al., 2020a). Extending our methods and evaluations to cover

multilingual sense embeddings is an important future direction.

7.3.2 Using Contextualised Word Embeddings as Source Embeddings

Our meta-sense embedding method described in Chapter 4 requires static sense embeddings, and cannot be applied to contextualised sense embedding methods such as SenseBERT (Levine et al., 2020). There has been some work on learning word-level and sentence-level (Takahashi and Bollegala, 2022; Poerner et al., 2020) meta-embeddings using contextualised word embeddings produced by MLMs as the source embeddings. However, contextualised sense embedding methods are limited compared to the numerous static sense embedding methods. This is partly due to the lack of large-scale sense annotated corpora, required to train or fine-tune contextualised sense embeddings. Extending our work to learn meta-sense embeddings using contextualised word embeddings as source embeddings is an interesting future research direction.

7.3.3 Extending the Dataset for Evaluating Social Biases in Sense Embeddings

Given that our SSSB dataset introduced in Chapter 5 is generated from a handful of manually written templates, it is far from complete. In addition, our gender-bias evaluation is limited to binary (male vs. female) genders and racial-bias evaluation is limited to Black as a race. Extending the categories is identified as important and necessary for future research directions. On the other hand, simply because a sense embedding does not show any social biases on SSSB according to the evaluation metrics we use in this thesis *does not* mean that it would be appropriate to deploy it in downstream NLP applications that require sense embeddings. In particular, task-specific fine-tuning of even

bias-free embeddings can result in novel unfair biases from creeping in. Therefore, developing debiasing methods for sense embeddings will be a natural line of future work.

Bibliography

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of COLING*, pages 1638–1649.

Felipe Almeida and Geraldo Xexéo. 2019. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*.

Jacob Andreas and Dan Klein. 2015. When and why are log-linear models self-normalizing? In *Proceedings of NAACL HLT*, pages 244–249, Denver, Colorado. Association for Computational Linguistics.

Alan Ansell, Felipe Bravo-Marquez, and Bernhard Pfahringer. 2019. An ELMo-inspired approach to SemDeep-5's Word-in-Context task. In *SemDeep-5*, pages 21–25.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A Latent Variable Model Approach to PMI-based Word Embeddings. *TACL*, 4:385–399.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *TACL*, 6:483–495.

Cong Bao and Danushka Bollegala. 2018. Learning word meta-embeddings by autoencoding. In *Proceedings of COLING*, pages 1650–1661.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021a. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of NAACL HLT*, pages 4661–4672.

Edoardo Barba, Luigi Procopio, Niccolo Campolungo, Tommaso Pasini, and Roberto Navigli. 2021b. Mulan: Multilingual label propagation for word sense disambiguation. In *Proceedings of IJCAI*, pages 3837–3844.

Roland Bauerschmidt, Tyler Helmuth, and Andrew Swan. 2021. The geometry of random walk isomorphism theorems. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 57(1).

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *JMLR*, 3:1137–1155.

Michele Bevilacqua and Roberto Navigli. 2019. Quasi Bidirectional Encoder Representations from Transformers for word sense disambiguation. In *Proceedings of RANLP*, pages 122–131.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of ACL*, pages 2854–2864.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of IJCAI*. International Joint Conference on Artificial Intelligence, Inc.

Sudha Bhingardive, Dhirendra Singh, V Rudramurthy, Hanumant Harichandra Redkar, and Pushpak Bhattacharyya. 2015. Unsupervised most frequent sense detection using word embeddings. In *HLT-NAACL*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR*, 3:993–1022.

Terra Blevins and Luke Zettlemoyer. 2020. Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. In *Proceedings of ACL*, pages 1006–1017.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *TACL*, 5:135–146.

Danushka Bollegala. 2022. Learning meta word embeddings by unsupervised weighted concatenation of source embeddings. In *Proceedings of IJCAI-ECAI*.

Danushka Bollegala and Cong Bao. 2018. Learning word meta-embeddings by autoencoding. In *Proceedings of COLING*, pages 1650–1661, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2018. Think Globally, Embed Locally — Locally Linear Meta-embedding of Words. In *Proceedings of IJCAI-EACI*, pages 3970–3976.

Danushka Bollegala and James O'Neill. 2022. A Survey on Word Meta-Embedding Learning. In *Proceedings of the IJCAI-ECAI*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of NeurIPS*, volume 29.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. In *Proceedings of EMNLP*, pages 615–620.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceddings of NeurIPS*, volume 33, pages 1877–1901.

Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard Zemel. 2019. Understanding the origins of bias in word embeddings. In *Proceedings of ICLR*, pages 803–811. PMLR.

Agostina Calabrese, Michele Bevilacqua, and Roberto Navigli. 2021. Evilbert: Learning task-agnostic multimodal sense embeddings. In *Proceedings of IJCAI-PRICAI*, pages 481–487.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017a. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017b. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356:183–186.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018a. From word to sense embeddings: A survey on vector representations of meaning. *JAIR*, 63:743–788.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018b. From Word To Sense Embeddings: A Survey on Vector Representations of Meaning. *JAIR*, 63:743–788.

Yee Seng Chan, Hwee Tou Ng, et al. 2005. Scaling up word sense disambiguation via parallel texts. In *Proceedings of AAAI*, volume 5, pages 1037–1042.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP*, pages 1025–1035.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *In proceedings of ICASSP*, pages 4774–4778. IEEE.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR*.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proceedings of NAACL-HLT*, pages 194–198, New Orleans, Louisiana. Association for Computational Linguistics.

Davide Colla, Enrico Mensa, and Daniele P Radicioni. 2020. Lesslex: Linking multilingual embeddings to SenSe representations of LEXical items. *Computational Linguistics*, 46(2):289–333.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of EMNLP*, pages 670–680.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proceedings of NeurIPS*, volume 32. Curran Associates, Inc.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. *Proceedings of NeurIPS*, 28.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of AAAI*, volume 32.

Sunipa Dev, Tao Li, Jeff Phillips, and Vivek Srikumar. 2020. On Measuring and Mitigating Biased Inferences of Word Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7659–7666.

Sunipa Dev and Jeff Phillips. 2019. Attenuating bias in word vectors. In *Proceedings of AIStat*, pages 879–887. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL HLT*, pages 4171–4186.

Simon De Deyne, Danielle J. Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2019. The "small world of words" english word association norms for over 12,000 cue words. *Behavior Research Methods*, 51(3):987–1006.

Paramveer Dhillon, Dean P Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via cca. In *Proceedings of NeurIPS*, volume 24.

Yupei Du, Yuanbin Wu, and Man Lan. 2019. Exploring Human Gender Stereotypes with Word Association Test. In *Proceedings of EMNLP-IJCNLP*, pages 6132–6142, Hong Kong, China. Association for Computational Linguistics.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Andreas Eisele and Yu Chen. 2010. Multiun: A multilingual corpus from united nation documents. In *Proceedings of LREC*.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. Understanding undesirable word embedding associations. In *Proceedings of ACL*, pages 1696–1705, Florence, Italy. Association for Computational Linguistics.

Christiane Fellbaum and George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of ICML*, volume 80, pages 1602–1611.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439.

Iker García, Rodrigo Agerri, and German Rigau. 2020. A Common Semantic Space for Monolingual and Cross-Lingual Meta-Embeddings. *arXiv preprint arXiv:2001.06381*.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *In Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations. In *Proceedings of EMNLP-IJCNLP*, pages 5300–5309.

Zellig Harris. 1954. Distributional Structure. *Word*, 10(23):146–162.

Tatsunori Hashimoto, David Alvarez-Melis, and Tommi Jaakkola. 2016. Word Embeddings as Metric Recovery in Semantic Spaces. *TACL*, 4:273–286.

Bradley Hauer, Yixing Luan, and Grzegorz Kondrak. 2019. You shall know the most frequent sense by the company it keeps. In *Proceedings of ICSC*, pages 208–215. IEEE.

Taher H. Haveliwala, Aristides Gionis, Dan Klein, and Piotr Indyk. 2002. Evaluating strategies for similarity search on the web. In *Proceedings of International World Wide Web Conference*.

Jingyi He, KC Tsiolis, Kian Kenyon-Dean, and Jackie Chi Kit Cheung. 2020a. Learning Efficient Task-Specific Meta-Embeddings with Word Prisms. In *Proc. of COLING*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020b. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Harold Hotelling. 1953. Canonical correlation analysis (cca). *Journal of Educational Psychology*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of ACL*, pages 328–339.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*, pages 873–882.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. In *EMNLP-IJCNLP*, pages 3500–3505.

Ngo Quang Huy, Tu Minh Phuong, and Ngo Xuan Bach. 2022. Autoencoding Language Model Based Ensemble Learning for Commonsense Validation and Explanation. *arXiv preprint arXiv:2204.03324*.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*, pages 95–105.

Pratik Jawanpuria, Satya Dev N T V, Anoop Kunchukuttan, and Bamdev Mishra. 2020. Learning Geometric Word Meta-Embeddings. In *Reps4NLP*, pages 39–44, Online.

John E Joseph. 2006. *Language and politics*. Edinburgh University Press.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of ACL*, pages 1641–1650, Florence, Italy.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. In *Proceedings of EACL*, pages 1256–1266, Online.

Masahiro Kaneko and Danushka Bollegala. 2022a. Unmasking the Mask – Evaluating Social Biases in Masked Language Models. In *Proceedings of AAAI*, volume 36, pages 11954–11962, Vancouver, BC, Canada.

Masahiro Kaneko and Danushka Bollegala. 2022b. Unmasking the mask–evaluating social biases in masked language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11954–11962.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic Meta-Embeddings for Improved Sentence Representations. In *Proceedings of EMNLP*, pages 1466–1477.

Josef Klafka and Allyson Ettinger. 2020. Spying on Your Neighbors: Fine-grained Probing of Contextual Embeddings for Information about Surrounding Words. In *Proceedings of ACL*, pages 4801–4811, Online. Association for Computational Linguistics.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of ACL*, pages 5670–5681.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of GeBNLP*, pages 166–172, Florence, Italy. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of EMNLP*, pages 785–794.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A Lite Bert for Self-supervised Learning of Language Representations. In *Proceedings of ICLR*.

Rémi Lebret and Ronan Collobert. 2014. Word Embeddings through Hellinger PCA. In *Proceedings of EACL*, CONF.

Guang-He Lee and Yun-Nung Chen. 2017. MUSE: Modularizing Unsupervised Sense Embeddings. In *Proceedings of EMNLP*, pages 327–337.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of ACL*, pages 4656–4667, Online. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NeurIPS*, volume 27.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3? In *Proceedings of DeeLIO*, pages 100–114.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Daniel Loureiro and Jose Camacho-Collados. 2020. Don't Neglect the Obvious: On the Role of Unambiguous Words in Word Sense Disambiguation. In *Proceedings of EMNLP*, pages 3514–3520.

Daniel Loureiro and Alipio Jorge. 2019a. Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy.

Daniel Loureiro and Alípio Mário Jorge. 2019b. LIAAD at Semdeep-5 Challenge: Word-in-context (WiC). In *Proceedings of SemDeep-5*.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208.

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the EMNLP*, pages 1402–1411.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *Proceedings of ACL*, pages 2473–2482.

John C Mallery. 1988. Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers. In *Master's thesis, MIT Political Science Department*. Citeseer.

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2017. Embedding Words and Senses Together via Joint Knowledge-Enhanced Training. In *Proceedings of CoNLL*, pages 100–111.

Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. In *Proceedings of NAACL-HLT*, pages 615–621, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. In *Proceedings of EMNLP-IJCNLP*, pages 3534–3540.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. In *Proceedings of NAACL-HLT*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Proceedings of NeurIPS*, volume 30.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 33(4):553 – 590.

Diana McCarthy, Rob Koeling, Julie Weeds, and John A Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of ACL*, pages 279–286.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*, volume 26.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceed-*

ings of SemEval 2015, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *TACL*, 2:231–244.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and Effective Postprocessing for Word Representations. In *Proceedings of ICLR*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of ACL-IJCNLP*, pages 5356–5371, Online. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings EMNLP*, pages 1953–1967, Online. Association for Computational Linguistics.

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

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of SemEval 2013*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

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

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple em-

beddings per word in vector space. In *Proceedings of EMNLP*, pages 1059–1069.

James O'Neill and Danushka Bollegala. 2018. Meta-embedding as auxiliary task regularization. In *Proceedings of EACI*.

Tommaso Pasini and Roberto Navigli. 2018. Two knowledge-based methods for high-performance sense distribution learning. In *Proceedings of AAAI*, volume 32.

Tommaso Pasini and Roberto Navigli. 2020. Train-o-matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279:103215.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. XL-WSD: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proceedings of AAAI*, volume 35, pages 13648–13656.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP*, pages 1532–1543.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of ACL*, pages 1756–1765.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019a. Knowledge Enhanced Contextual Word Representations. In *EMNLP-IJCNLP*, pages 43–54.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019b. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of RepL4NLP*, pages 7–14.

Tommaso Petrolito and Francis Bond. 2014. A survey of wordnet annotated corpora. In *Proceedings of the Global WordNet Conference*, pages 236–245.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of NAACL-HLT*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-Conflated Semantic Representations. In *Proceedings EMNLP*, pages 1680–1690.

Mohammad Taher Pilehvar and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. Sentence Meta-Embeddings for Unsupervised Semantic Textual Similarity. In *Proceedings of ACL*, pages 7027–7034.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval-2007*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI publications*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI publications*.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of EACL*, pages 99–110.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization. In *Proceedings of EMNLP*, pages 7193–7206, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of EMNLP*, pages 2383–2392.

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 ACL*, pages 7237–7256. Association for Computational Linguistics.

Joseph Reisinger and Raymond Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of NAACL*, pages 109–117.

Douglas LT Rohde, Laura M Gonnerman, and David C Plaut. 2006. An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8(627-633):116.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In *Proceedings of ACL-IJCNLP*, pages 1793–1803.

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*.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2019. Just "OneSeC" for Producing Multilingual Sense-Annotated Data. In *Proceedings of ACL*, pages 699–709.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020a. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proceedings of AAAI*, volume 34, pages 8758–8765.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of EMNLP*, pages 3528–3539, Online.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Computing Research Repository*, arXiv:2103.00453.

Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized PageRank with Syntagmatic Information for Multilingual Word Sense Disambiguation. In *Proceedings of ACL*, pages 37–46.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of ACL*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive Biases in Natural Language Processing Models: A Conceptual

Framework and Overview. In *Proceedings of ACL*, pages 5248–5264. Association for Computational Linguistics.

Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. 2019. Retrofitting contextualized word embeddings with paraphrases. In *Proceedings of EMNLP-IJCNLP*, pages 1198–1203, Hong Kong, China. Association for Computational Linguistics.

Andrew Silva, Pradyumna Tambwekar, and Matthew Gombolay. 2021. Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers. In *Proceedings of NAACL-HLT*, pages 2383–2389, Online. Association for Computational Linguistics.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL-3*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of CoNLL*, pages 338–344.

Keigo Takahashi and Danushka Bollegala. 2022. Unsupervised attention-based sentence-level meta-embeddings from contextualised language models. In *Proceedings of LREC*.

Wilson L. Taylor. 1953. Wordnet: a lexical database for english. *Journalism Bulletin*, 30(4):415–433.

François Torregrossa, Robin Allesiardo, Vincent Claveau, Nihel Kooli, and Guillaume Gravier. 2021. A survey on training and evaluation of word embeddings. *IJDSA*, 11(2):85–103.

Rocco Tripodi and Roberto Navigli. 2019. Game theory meets embeddings: a unified framework for word sense disambiguation. In *Proceedings of EMNLP-IJCNLP*, pages 88–99.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *JAIR*, 37:141–188.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*, volume 30.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference*, pages 108–117.

Thuy Vu and D Stott Parker. 2016. K-Embeddings: Learning Conceptual Embeddings for Words using Context-Embeddings: Learning Conceptual Embeddings for Words using Context. In *Proceedings of NAACL*, pages 1262–1267.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of NeurIPS*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of BlackboxNLP*, pages 353–355.

Ming Wang and Yinglin Wang. 2020. A synset relation-enhanced framework with a try-again mechanism for word sense disambiguation. In *Proceedings of EMNLP*, pages 6229–6240.

Tianlu Wang, Xi Victoria Lin, Nazneen Fatema Rajani, Bryan McCann, Vicente Ordonez, and Caiming Xiong. 2020. Double-hard debias: Tailoring word embeddings for gender bias mitigation. In *Proceedings of ACL*, Online. Association for Computational Linguistics.

Warren Weaver. 1945. Translation. *Machine Translation of Languages: Fourteen Essays*.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of ACL-IJCNLP*, pages 323–333.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

Genta Indra Winata, Zhaojiang Lin, and Pascale Fung. 2019a. Learning multilingual meta-embeddings for code-switching named entity recognition. In *Proceedings of RepL4NLP*, pages 181–186.

Genta Indra Winata, Zhaojiang Lin, Jamin Shin, Zihan Liu, and Pascale Fung. 2019b. Hierarchical meta-embeddings for code-switching named entity recognition. *arXiv preprint arXiv:1909.08504*.

Xin Wu, Yi Cai, Yang Kai, Tao Wang, and Qing Li. 2020. Task-oriented domain-specific meta-embedding for text classification. In *Proceedings of EMNLP*, pages 3508–3513.

Yuqiang Xie, Yue Hu, Luxi Xing, and Xiangpeng Wei. 2019. Dynamic task-specific factors for meta-embedding. In *Proceedings of KSEM*, pages 63–74. Springer.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*, volume 32.

Boon Peng Yap, Andrew Koh, and Eng Siong Chng. 2020. Adapting bert for word sense disambiguation with gloss selection objective and example sentences. In *Findings of EMNLP*, pages 41–46.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of ACL*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Zi Yin and Yuanyuan Shen. 2018. On the Dimensionality of Word Embedding. In *Proceedings of NeurIPS*, pages 887–898.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018a. Learning gender-neutral word embeddings. In *Proceedings of EMNLP*, pages 4847–4853.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. Learning Gender-Neutral Word Embeddings. In *Proceedings of EMNLP*, pages 4847–4853, Brussels, Belgium.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *Proceedings of NeurIPS*, volume 16. MIT Press.

Yi Zhou and Danushka Bollegala. 2021. Learning Sense-Specific Static Embeddings using Contextualised Word Embeddings as a Proxy. In *Proceedings of PACLIC*, pages 11–20, Shanghai, China. Association for Computational Linguistics.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*, pages 1393–1398.