LMMS reloaded: Transformer-based sense embeddings for disambiguation and beyond

Daniel Loureiro [a,*], Alípio Mário Jorge [a], Jose Camacho-Collados [b]

[a] *LIAAD - INESC TEC, Dept. of Computer Science, FCUP, University of Porto, Portugal*
[b] *School of Computer Science and Informatics, Cardiff University, United Kingdom*

ARTICLE INFO

ABSTRACT

Distributional semantics based on neural approaches is a cornerstone of Natural Language Processing, with surprising connections to human meaning representation as well. Recent Transformer-based Language Models have proven capable of producing contextual word representations that reliably convey sense-specific information, simply as a product of self-supervision. Prior work has shown that these contextual representations can be used to accurately represent large sense inventories as sense embeddings, to the extent that a distance-based solution to Word Sense Disambiguation (WSD) tasks outperforms models trained specifically for the task. Still, there remains much to understand on how to use these Neural Language Models (NLMs) to produce sense embeddings that can better harness each NLM's meaning representation abilities. In this work we introduce a more principled approach to leverage information from all layers of NLMs, informed by a probing analysis on 14 NLM variants. We also emphasize the versatility of these sense embeddings in contrast to task-specific models, applying them on several sense-related tasks, besides WSD, while demonstrating improved performance using our proposed approach over prior work focused on sense embeddings. Finally, we discuss unexpected findings regarding layer and model performance variations, and potential applications for downstream tasks.

1. Introduction

Lexical ambiguity is prevalent across different languages and plays an important role in improving communication efficiency [87]. Word Sense Disambiguation (WSD) is a long-standing challenge in the field of Natural Language Processing (NLP), and Artificial Intelligence more generally, with an extended history of research in computational linguistics [75].

Interestingly, both computational and psychological accounts of meaning representation have converged on high-dimensional vectors within semantic spaces.

From the computational perspective, there is a rich line of work on learning word embeddings based on statistical regularities from unlabeled corpora, following the well-established Distributional Hypothesis [41,35, DH]. The first type of distributional word representations relied on count-based methods, initially popularized by LSA [27], and later refined with GloVe [82]. Before GloVe, word embeddings learned with neural networks, first introduced by Bengio et al. [7], gained wide adoption with word2vec [72] and, afterwards, culminated with fastText [12]. The development and improvement of word embeddings has been a major contributor to the progress of NLP in the last decade [38].

---

* Corresponding author.
E-mail addresses: daniel.b.loureiro@inesctec.pt (D. Loureiro), amjorge@fc.up.pt (A. Mário Jorge), camachocolladosj@cardiff.ac.uk (J. Camacho-Collados).

From the psychological perspective, there is also ample behavioral evidence in support of distributional representations of word meaning. Similarly to word embeddings, these representations are related according to the degree of shared features within semantic spaces, which translates into proximity in vector-space [97,48]. Understandably, the nature of the features making up this psychological account of semantic space, among other aspects (e.g., learning method), is not as clear as we find in the computational account. Nevertheless, contextual co-occurrence is among the most informative factors for meaning representation as well [67,32,91]. There are even use cases in neurobiology motivating research into accurate distributional representations of word meaning. In Pereira et al. [83], word embeddings have proven useful for decoding words and sentences from brain activity, after learning a mapping between corpus-based embeddings (i.e., GloVe and word2vec) and fMRI activation.

The current understanding of how humans perform disambiguation attributes major relevance to sentential context, and other linguistic and paralinguistic cue's (e.g., speaker accent) to a lesser extent [97,14]. However, the previously mentioned computational approaches are not designed for sense-level representation due to the Meaning Conflation Deficiency [15], as they converge different senses into the same word-level representation. Some works have explored variations on the word2vec method for sense-level embeddings [99,45,90,65], but the dynamic word-level interactions composing sentential context were not targeted by those works.

The works of Melamud et al. [68], Yuan et al. [126], Peters et al. [84] were among the first to propose Neural Language Models (NLMs) featuring dynamic word embeddings conditioned on sentential context (i.e., contextual embeddings). These works showed that NLMs (trained exclusively on language modeling objectives) can produce contextual embeddings for word forms that are sensitive to the word's usage in particular sentences. Furthermore, these works also addressed WSD tasks with a simple nearest neighbors solution (k-NN) based on proximity between contextual embeddings. Their results rivaled systems trained specifically for WSD (i.e., with additional modeling objectives), highlighting the accuracy of these contextual embeddings.

However, it was not until the development of Transformer-based NLMs, namely BERT [29], that contextual embeddings from NLMs showed clearly better performance on WSD tasks than previous systems trained specifically for WSD (LMMS, Loureiro & Jorge [61]).

In this earlier work, we explored how to further take advantage of the representational power of NLMs through propagation strategies and encoding sense definitions. Besides pushing the state-of-the-art of WSD, in Loureiro & Jorge [61] we created sense embeddings for every entry in the Princeton WordNet v3.0 (200k word senses, [34]), so that the semantic space being represented is granular and expansive enough to encompass general knowledge domains for various parts-of-speech of the English language. With this fully populated semantic space at our disposal we suggested strategies for uncovering biases and world knowledge represented by NLMs.

Since our work on LMMS, others have shown additional performance gains for WSD with fine-tuning or classification approaches that make better usage of sense definitions [44,11], semantic relations from external resources [104,9], or altogether different approaches to WSD [5].

However, there are several questions still standing regarding how to leverage NLMs for creating accurate and versatile sense embeddings, beyond optimizing for WSD benchmarks only. Given that semantic spaces with distributional representations of word meanings feature prominently in both the conventional computational and psychological accounts of word disambiguation, these questions warrant further exploration.

Contributions. In this extension of LMMS, we broaden our scope to more recent Transformer-based models in addition to BERT [124,59,52] (14 model variants in total), verify whether they exhibit similar proficiency at sense representation, and explore how performance variation can be attributed to particular differences in these models. Striving for a principled approach to sense representation with NLMs, we also introduce a new layer pooling method, inspired by recent findings of layer specialization [95], which we show is crucial to effectively use these new NLMs for sense representation. Most importantly, in this article we provide a general framework for learning sense embeddings with Transformers and perform an extensive evaluation of such sense embeddings from different NLMs on various sense-related tasks, emphasizing the versatility of these representations.

Outline. This work is organized as follows. We first provide some background information on the main topics of this research: Vector Semantics (§2.1), Neural Language Modeling (§2.2) and Sense Inventories (§2.3). Next, we describe related work on Sense Embeddings (§3.1), WSD (§3.2) and Probing NLMs (§3.3).

The method used to produce this work's sense embeddings is described in Section 4, covering aspects of the method introduced in Loureiro & Jorge [61] (from §4.1 to §4.3), as well as our new layer pooling method in Section 4.4.

In Section 5 we describe our experimental setting, providing relevant details about our choice of NLMs (§5.1) and annotated corpora used to learn sense embeddings (§5.2).

The layer pooling methodology described in Section 4.4 requires validating performance under two distinct modes of application. Consequently, in Section 6 we report on performance variation per layer across NLMs (§6.1), highlight differences between disambiguation and matching profiles (§6.2), and present the rationale for choosing particular profiles for each task (§6.3).

In Section 7, we tackle several sense-related tasks using our proposed sense embeddings and compare results against the state-of-the-art, namely: WSD (§7.1), Uninformed Sense Matching (§7.2), Word-in-Context (§7.3), Graded Word Similarity in Context (§7.4) and Paired Sense Similarity (§7.5).

In order to better understand the contributions of this work, Section 8 reports on several ablation analyses targeting the following: choice of Sense Profiles (§8.1), impact of unambiguous word annotations (§8.2), merging gloss representations (§8.3), and indirect representation of synsets (§8.4).

We discuss our findings in Section 9, regarding representations from intermediate layers of NLMs (§9.1), irregularities across models and variants (§9.2), and potential downstream applications of our sense embeddings focusing on knowledge integration (§9.3).

Finally, in Section 10 we present our concluding remarks, and provide details about our release of sense embeddings, code and more.

2. Preliminaries

This work exploits the interaction between vector-based semantic representations (§2.1), recent developments on NLMs (§2.2), and curated sense inventories (§2.3). In this section we provide some background on these topics.

2.1. Vector semantics

Nearly a century ago, Firth [36] postulated that "the meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously". Indeed, after working on formal theories of word meaning definition, Wittgenstein [121] conceded "the meaning of a word is its use in a language". This view of meaning representation became known as the Distributional Hypothesis (DH) [41], which proposes that words that occur in the same contexts tend to have similar meanings. During this period, Osgood et al. [78] further proposed representing the meaning of words as points in multi-dimensional space, with similar words having similar representations, thus being placed closely in this space. Still, it would take a few more decades of computing advancements to appreciate the implications of the DH.

Early VSMs. After some early works introducing vector space models (VSMs) for information retrieval [102,103], Deerwester [28,27] was the first to use dense vectors to represent word meaning, initially with a method called Latent Semantic Indexing (LSI), and later with Latent Semantic Analysis (LSA). LSA was based on a word-document weighted frequency matrix from which the first 300-dimensions resulting from Singular-Value Decomposition (SVD) would correspond to word embeddings. Lund & Burgess [64] introduced another influential method similar to LSA, called Hyperspace Analogue to Language (HAL) which differed from LSA by considering word-word frequencies instead, introducing the notion of a fixed-sized window as context (e.g., the two words to the left and to the right) instead full documents, which would become the standard representation of context. Following these developments, Landauer & Dumais [53] evaluated the performance of LSA embeddings learned from large corpora on a simple semantic task (synonymy tests) and found that these embeddings performed comparably to school-aged children, when measuring similarity between word pairs as the cosine similarity between their corresponding embeddings (inspired by applications for information retrieval). Already in this early period, Schutze [107] and Yarowsky [125] realized the potential for WSD applications based on the similarity between unsupervised word embeddings. Blei et al. [10] would later introduce Latent Dirichlet Allocation (LDA) which uses a generative probabilistic approach to generalize and improve on the approach used for LSA, being widely adopted for topic modeling and other applications beyond semantic analysis.

Neural models. Having established that corpus-based word embeddings are able to capture semantic knowledge, additional progress followed swiftly. A milestone in the evolution of word embeddings was the discovery that Neural Language Models (NLMs) implicitly develop word embeddings when training for the task of word prediction [8]. Shortly after, Collobert [23–25] demonstrated that word embeddings could be incorporated into neural architectures for various NLP tasks. With word2vec, Mikolov et al. [73] distilled the components of NLMs responsible for learning word embeddings into a lightweight and scalable solution, allowing this neural-based solution to be employed on corpora of unprecedented size (100B tokens). Nevertheless, count-based solutions would still remain important, particularly GloVe [82], as these methods were also significantly improved. The next major improvement was the introduction of fastText [12], which was able to represent words absent from training data by leveraging subword information, as well as refining several aspects of word2vec's training method.

Sense embeddings. In spite of their success, word2vec, GloVe and fastText conflated different senses of the same word form into the same representation, a shortcoming known as the Meaning Conflation Deficiency [15]. While a number of extensions were proposed for the creation of sense-specific representations, such as AutoExtend [99], NASARI [18], DeConf [90] or Probabilistic FastText [4], this issue would require the development of a new generation of NLMs in order to be effectively addressed.

2.2. Neural language modeling

The first major step towards contextual embeddings from NLMs, was the development of context2vec [68], a single-layer bidirectional LSTM trained with the objective of maximizing similarity between hidden states and target word embeddings, similarly to word2vec. Peters et al. [84] built upon context2vec with ELMo, a deeper bidirectional LSTM trained with language modeling objectives that produce more transferrable representations. Both context2vec and ELMo emphasized WSD

applications, providing the most convincing accounts until then that sense embeddings can be effectively represented as centroids of contextual embeddings, showing 1-NN solutions to WSD tasks that rivaled the performance of task-specific models.

With the introduction of highly-scalable Transformer architectures [113], two kinds of very deep NLMs emerged: causal (or left-to-right) models, epitomized by the Generative Pre-trained Transformer [13, GPT-3], where the objective is to predict the next word given a past sequence of words; and masked models, where the objective is to predict a masked (i.e., hidden) word given its surrounding words, of which the most prominent example is the Bidirectional Encoder Representations from Transformers (BERT) [29]. The difference in training objectives results in these two varieties of NLMs specializing at different tasks, with causal models excelling at language generation and masked models at language understanding.[1]

BERT proved highly successfully at most NLP tasks [98], and motivated the development of numerous derivative models, many of which we also explore in this work. In spite of this progress, Transformer-based NLMs can still show strong reliance on surface features [66] and social biases which are hard to correct [127]. There are known theoretical limits to how much language understanding can be expected from models trained with language modeling objectives alone [6,69], and it is not clear how far current models are from those limits.

2.3. Sense inventories

The currently most popular English word sense inventory is the Princeton WordNet [34] (henceforth, WordNet), a large semantic network comprised of general domain concepts curated by experts.[2]

The core unit of WordNet is the synset, which represents a cognitive concept. Each lemma (word or multi-word expression) in WordNet belongs to one or more synsets, and word senses amount to the combination of word forms and synsets (referred as sensekeys). As a result, the set of words that belong to a synset can be described as synonyms, with some words being ambiguous (belonging to additional synsets) while others not (specific to a synset). The predominant semantic relation in WordNet, which relates synset pairs, is hypernymy (i.e., Is-A). Each synset also features a gloss (dictionary definition), part-of-speech (noun, verb, adjective or adverb) and lexname,[3] which is a syntactic category and logical grouping. Synsets are formally represented as numerical codes. Following related works, we also represent them using the more readable format $lemma_{POS}^{\#}$, where $lemma$ corresponds to synset's most representative lemma.

As an example, the lemma 'mouse' is polysemous belonging to the $mouse_n^1$ (rodent) and $mouse_n^4$ (computer mouse) synsets, among others. The most frequent sense for mouse, mouse%1:05:00:: (sensekey), belongs to the synset $mouse_n^1$ (02330245n) which has an hypernymy relation with $rodent_n^1$, lexname 'noun.animal', and gloss "any of numerous small rodents typically [...]".

Following Loureiro & Jorge [61], we use WordNet version 3.0, which contains 117,659 synsets, 206,949 senses, 147,306 lemmas, and 45 lexnames.

3. Related work

In this section we cover related work on the various well-researched topics that our work intersects, namely Sense Embeddings (§3.1), WSD (§3.2) and Probing NLMs (§3.3).

3.1. Sense embeddings

Sense embeddings emerged in NLP due to the so-called meaning conflation deficiency of word embeddings [15]. By merging several meanings into a single representation, the single vector proved insufficient in certain settings [123], and contradicted common laws in distance metrics, such as the triangle inequality [77]. In order to solve this issue, the field of sense vector representation mainly split into two categories: (1) unsupervised, where senses were learned directly from text corpora [96,43,118]; (2) or knowledge-based, where senses were linked to a pre-defined sense inventory by exploiting an underlying knowledge resource [99,90,65,20].

In this article, we focus on the latter type of representation, particularly leveraging powerful Transformer-based language models trained on unlabeled text corpora. As such, the final representation is mainly constructed based on the knowledge learned by the language models, and knowledge resources such as WordNet serve to guide the annotation process. The goal of this paper is indeed to construct a task-agnostic sense representation that can be leveraged in semantic and textual applications. This differs from traditional static sense embeddings which, with a few notable exceptions [56,37,89], were mainly leveraged in intrinsic sense-based tasks only. As we show throughout this paper, general-purpose sense representations learned with the power of Transformers and guided through an underlying lexical resource such as WordNet prove to be robust in a range of text-based semantic tasks, as well as in intrinsic sense-based benchmarks.

---

[1] Although recent models like BART [55] show progress towards both.
[2] Babelnet [76], Wiktionary [70] and HowNet [31] are popular alternatives covering other languages.
[3] Lexnames are also known as supersenses [37,89].

3.2. Word sense disambiguation

As one of the earliest Artificial Intelligence tasks, WSD has a long history of research. In this work, our coverage of related work for WSD is focused on recent systems using Transformer-based architectures for two reasons: our own experiments are also focused on Transformer-based systems; the current state-of-the-art for WSD has converged on these systems. Additionally, we also distinguish between solutions addressing WSD from the nearest neighbors paradigm, using pre-computed sense embeddings, and task-specific solutions fine-tuning Transformer models or training classifiers using their internal representations.

3.2.1. Nearest neighbors

Our prior LMMS work (described throughout this paper) was the first to demonstrate that a nearest neighbors solution based on sense embeddings pooled from internal representations of BERT (i.e., feature extraction) could clearly outperform the state-of-the-art of the time, which still had not adopted Transformer-based models.

SensEmBERT [104] followed a similar approach to LMMS, but leveraged BabelNet to reduce dependency on annotated corpora, producing sense embeddings that performed better on WSD, though limited to nouns only.

With ARES, Scarlini et al. [105] introduce a method to produce a large number of semi-supervised annotations to dramatically increase the coverage of the sense inventory, and demonstrated that sense embeddings learned from those annotations can perform substantially better on WSD than LMMS.

SensEmBERT and ARES use the same layer pooling method and gloss embeddings as LMMS, although both have employed not only BERT-L, but also its multilingual variant, showing strong performance on languages other than English as well.

In addition to WSD, to our knowledge, the only other task these works have applied their sense embeddings is Word-in-Context (WiC) [88], which we also address in this work.

3.2.2. Trained classifiers

When it comes to using Transformers to train classifiers specific to the WSD task, we encounter a much more diverse set of solutions in comparison to feature extraction approaches.

One of the earliest and most straightforward supervised classifiers for WSD using BERT was the Sense Vocabulary Compression (SVC) of Vial et al. [115], which added layers to BERT, topped with a softmax classifier, to be trained targeting a strategically reduced set of admissible candidate senses.

Following outstanding results on a range of text classification tasks by model fine-tuning, GlossBERT [44] fine-tuned BERT using glosses so that WSD could be framed as a text classification task pairing glosses to words in context. KnowBERT [86] employs a more sophisticated fine-tuning approach, designed to exploit knowledge bases (WordNet and Wikipedia) as well as glosses.

Straying further from prototypical classifiers, Blevins & Zettlemoyer [11] (BEM) propose a bi-encoder method which learns to represent senses based on glosses while performing the optimization jointly with the underlying BERT model. Taking advantage of an ensemble of sense embeddings from LMMS and SensEmBERT, along with additional resources, EWISER [9] trains a multifaceted high performance WSD classifier.

Finally, the current state-of-the-art for WSD is ConSeC [5], which obtains impressively strong results by framing WSD as an extractive task, similar to extractive question answering, trained through fine-tuning BART [55], a sequence-to-sequence Transformer which outperforms BERT on reading comprehension tasks (while being of comparable size).

In Loureiro et al. [63] we extensively compared fine-tuning and feature extraction approaches for the WSD task. Consistent with prior work, we found that fine-tuning overall outperforms feature extraction. However, under comparable circumstances, the performance gap is narrow and feature extraction shows improved few-shot performance and less frequency bias.

3.3. Probing neural language models

As NLMs became popular, investigating properties of their internal states, or intermediate representations, also became an important line of research, often referred to as 'model probing'. Probing operates under the assumption that if a relatively simple classifier, based exclusively on representations from NLMs, can perform well at some task, then the required information was already encoded in the representations. For clarity, we define probes as functions (learned or heuristic) designed to reveal some intrinsic property of NLMs. In this section we cover probing works focused on lexical semantics and layer-specific variation that inspired our probing analysis. We distinguish these works by their use of probes trained using representations (learned), and probes directly comparing or analyzing unaltered representations (heuristics, such as nearest neighbors).

3.3.1. Learned probes

Among the most influential findings in this line of research was the discovery by Hewitt & Manning [42] that syntactically valid parse trees could be uncovered from linear transformations of word representations obtained from pre-trained ELMo and BERT models. Motivated by this discovery, Reif et al. [95] performed additional experiments focused on sense representation, including showing that a nearest neighbors based on BERT representations could outperform the reported

WSD state-of-the-art, particularly when following Hewitt & Manning [42]'s methodology to learn a probe tailored to sense representation. To increase sensitivity to sense-specific information, Reif et al. [95] used a loss that considered the difference between the average cosine similarity of embeddings of words with the same senses, and embeddings of words with different senses. Both Hewitt & Manning [42] and Reif et al. [95] approaches are designed for probing representations obtained from single layers.

With ELMo, Peters et al. [84] introduced contextualized word representations that are obtained from a linear combination of representations from all layers of the model. This linear combination uses task-specific weights learned through an optimization process, often referred to in the literature as "scalar mixing", and produced better results in downstream tasks when compared to representations obtained from individual layers. On closer inspection, Peters et al. [85] concluded that top layers can be less effective for semantic tasks possibly due to specialization for the language modeling tasks optimized during pre-training.

Tenney et al. [112] proposed an "edge probing" methodology, using scalar mixing, that allowed for evaluating different syntactic or semantic properties using a common classifier architecture, where probing models are trained to predict graph edges independently. In Tenney et al. [111], edge probing was employed to reveal that BERT implicitly performed different steps of a traditional NLP pipeline, in the expected order as information flows through the model, with lower layers processing local syntax (e.g., Part-of-Speech) and higher layers processing complex semantics of arbitrary distance (e.g., Semantic Roles). Raising concerns about remaining faithful to the information encoded in the representations, Kuznetsov & Gurevych [51] proposes reducing the expressive power of learned probes while improving edge probing.

Liu et al. [58] ran several probing experiments with simpler probes (i.e., linear classifiers), investigating differences between NLM architectures, namely ELMo, GPT and BERT, while still finding competitive performance with state-of-the-art task-specific models. They confirm that LSTM-based models (i.e., ELMo) present more task-specific (less transferable) top layers, but Transformers-based models (i.e., BERT) are less predictable and do not exhibit monotonic increase in task-specificity, in line with our own findings. GPT was found to significantly underperform ELMo and BERT, which Liu et al. [58] attributes to the fact that GPT is trained unidirectionally (left-to-right), while ELMo and BERT are trained bidirectionally.

3.3.2. Representational similarity

Without recourse to learned probes, Ethayarajh [33] investigated differences between ELMo, GPT-2 and BERT, relying on experiments based on cosine similarity to learn about the context-specificity of their representations. Ethayarajh [33] found that top layers show highest degree of context-specificity, but all layers of all three models produced highly anisotropic representations, with directions in vector space confined to a narrow cone, concluding that this property is an inherent consequence of the contextualization process. The anisotropy observed for all contextualized NLMs also supports the hypothesis of Reif et al. [95] that sense-level information is encoded in a low-dimensional subspace, since contextualization is crucial for sense disambiguation.

Vulić et al. [119] reached similar conclusions regarding the detrimental contribution of top layers for lexical tasks (e.g., lexical semantic similarity) while also finding improved results from averaging different layers, particularly task-specific layer subsets, prompting further research into layer weighting or meta-embedding approaches, and motivating the present work. Through direct comparison of cosine similarities, Chronis & Erk [19] reached similar conclusions as Vulić et al. [119] about the role of top layers for lexical similarity tasks, adding that top layers appear to better approximate relatedness than similarity.

Voita et al. [116] probed Transformer-based NLMs from an Information-Bottleneck perspective to learn about differences in information flow across the network according to language modeling pre-training objectives, particularly left-to-right, MLM, and translation. They find that the MLM objective induces representation of token identity in the lower layers, followed by a more generalized token representation in intermediate layers, and then token identity information gets recreated at top layers.

Mickus et al. [71] specifically verified whether BERT representations comprise a coherent semantic space. These experiments are explicitly detached from learned probes, as Mickus et al. [71] explains that such methodology interferes with direct assessment of the coherence of the semantic space as produced by NLMs. Using cluster analyses, they find that BERT indeed appears to represent a coherent semantic space (based only on representations from the final layer), although its Next Sentence Prediction (NSP) modeling objective leads to encoding semantically irrelevant information (sentence position), corrupting similarity relationships and complicating comparisons with other NLMs.

4. Method

We propose a principled approach for sense representation based on contextual NLMs trained exclusively with self-supervision. This approach is an extension of Loureiro & Jorge [61], addressing relevant issues still largely unresolved, particularly the influence of embeddings from the different layers composing NLMs, with the introduction of a novel layer probing methodology. Moreover, in this work, we reinforce the distinction between sense disambiguation and sense matching by introducing methodological differences specific to each application scenario.

This section starts by explaining the methods used in Loureiro & Jorge [61] for learning (§4.1), extending (§4.2) and applying sense embeddings (§4.3). Afterwards, we introduce our proposed layer probing methodology (§4.4), including how the resulting analysis informs a grounded pooling operation for combining embeddings from all layers of a NLM.
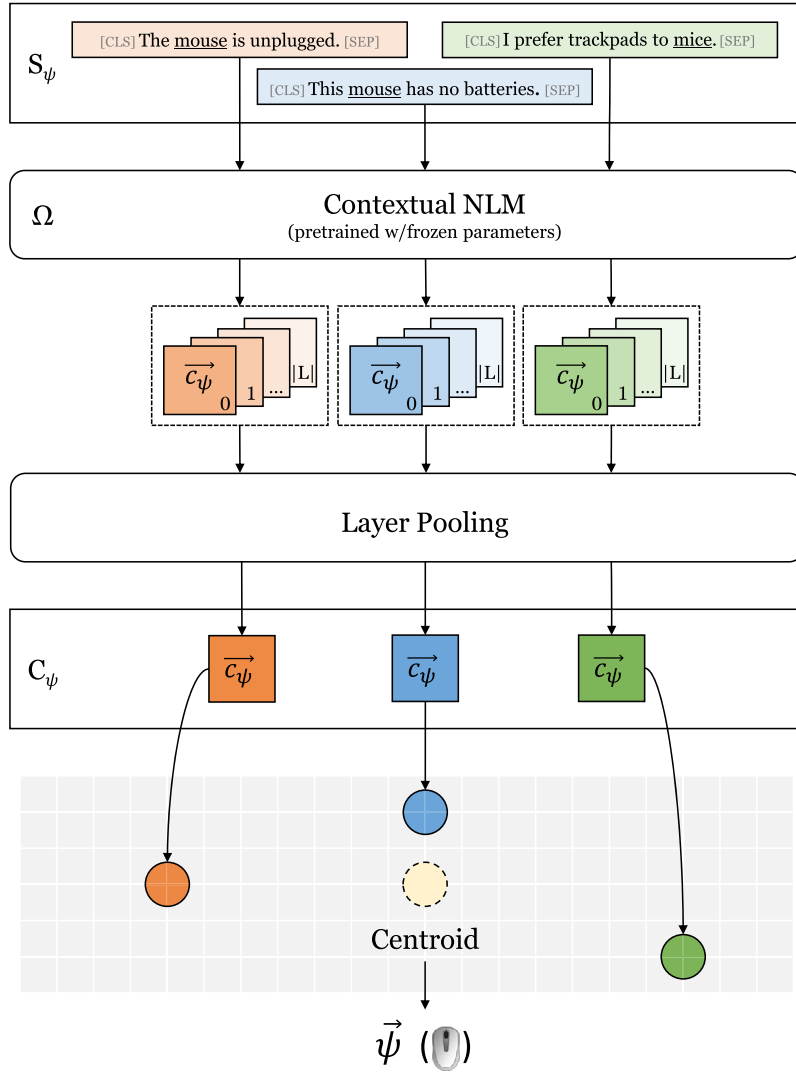
Fig. 1. Overview of learning sense embeddings from annotated corpora. Showing how the sense ψ for 'computer mouse' is determined from a set for sentences annotated with that sense S_ψ (padded with special tokens as expected by the NLM Ω). After pooling contextual embeddings C_ψ from layers L, the sense embedding for $\vec{\psi}$ is computed as the centroid of C_ψ.

4.1. Learning sense embeddings

The initial process to learn sense embeddings is based on sense-annotated sentences and contextualized embeddings of annotated words or phrases in context. An overview of the process can be seen at Fig. 1.

Formally, in order to generate sense embeddings learned in context from natural language, we require a pre-trained contextual NLM Ω (frozen parameters) and a corpus of sense-annotated sentences S. Every sense ψ is represented from the set of contextual embeddings $\vec{c}_l \in C_\psi$, obtained by employing Ω on the set of sentences S_ψ annotated with that sense (considering only contextual embeddings specific to tokens annotated with sense ψ), using representations at each layer $l \in L$, such that:

$$\vec{\psi} = \frac{1}{|C_\psi|} \sum_{l \in L} \sum_{\vec{c} \in C_\psi} \vec{c}_l \, , \, where \, C_\psi = \Omega(S_\psi) \tag{1}$$

The L set of layers typically used for sense representation is the last four $[-1, -2, -3, -4]$ (reversed layer indices), as discussed in Section 3.1.

Contextual NLMs typically operate at the subword-level, so the token-level embeddings $\vec{c}$ produced by Ω correspond to the average of each token's subword contextual embeddings (depending on the NLM, these may be BPE or WordPiece embeddings). Similarly, whenever sense-annotations cover a span of several tokens, we also use the average of the corre-

sponding token-level embeddings as the contextual embedding. Contextual NLMs are pre-trained using special tokens at specific locations, so we also include these tokens in their expected positions (e.g., `[CLS]` at the start and `[SEP]` at the end with BERT).

As described in Section 2.3, WordNet can be used to represent senses in two ways: sensekeys and synsets. Sense-annotated corpora most often use sensekey annotations, so in those cases sensekey embeddings do not require any intermediate mapping. Synset embeddings can be derived from sensekey annotations in at least two ways, which we differentiate as 'direct' and 'indirect'. In the direct approach, each sensekey annotation is converted (mapped) to the corresponding synset, so synset representations are learned from each annotation instance. In the indirect approach we first learn sensekey-level embeddings, without converting annotations, and afterwards compute synset embeddings as the average of corresponding sensekey embeddings. The latter approach has been explored in earlier works in sense embeddings [99]. In this work we explore both approaches.

4.2. Extending coverage with additional resources

Given that one of the major issues in supervised WSD is the lack of sense annotations [80], not just in their quantity but also in terms of their coverage of the sense inventory, we require solutions to address this in our method. On this matter, we also follow the methods we first proposed in Loureiro & Jorge [61] and later optimized with the introduction of the UWA corpus in Loureiro & Camacho-Collados [60]. The two methods, ontological propagation and gloss representation, are designed to reach full coverage of the sense inventory, and they are complementary by exploiting different resources, namely semantic relations between senses and glosses (combined with lemmas).

4.2.1. Ontological propagation

In Section 2.3 we introduced WordNet and the different elements and relations composing this semantic network. The ontological propagation method we presented in Loureiro & Jorge [61] exploits these relations between senses in WordNet in order to infer embeddings for senses which may not occur in annotated corpora. It is possible to infer accurate sense embeddings from these relations due to the fine-granularity of WordNet, along with widespread synonymy and hypernymy relations, to the extent that in Loureiro & Camacho-Collados [60] we showed that even annotations for unambiguous words can significantly improve the propagation process.

Since available corpora do not provide full-coverage annotations for our sense inventory of interest, by following the process described in Section 4.1 we are left with a represented senses Ψ, and a set of unrepresented senses Ψ'. The propagation process involves three steps, using increasingly abstract relations from WordNet - sets of synonyms (synsets), hypernymy relations, and lexical categories (lexnames or supersenses). In case we are targeting synset-level representations, then the first step/level should be skipped.

Considering we are provided mappings between sensekeys, synsets, hypernyms and lexnames, we infer Ψ' iteratively following Algorithm 1. After each of these sequential steps, every inferred $\vec{\psi}$ is added to the set of represented senses Ψ. This propagation method ensures full-coverage provided that initial sense embeddings Ψ are sufficiently diverse such that falling back on propagating from lexnames (supersenses) is always possible.

Algorithm 1: Propagation method to infer unrepresented senses Ψ', using sense embeddings Ψ learned from annotations, and relations R.

$\textbf{Propagate } (\Psi, \Psi', R)$
    $\textbf{foreach } \textit{unrepresented sense } \psi' \in \Psi' \textbf{ do}$
        $R_{\psi'} \leftarrow \{\text{all represented } \vec{\psi} \in \Psi \text{ for which } (\psi, \psi') \in R\};$
        $\textbf{if } |R_{\psi'}| > 0 \textbf{ then}$
            $\vec{\psi}' \leftarrow \text{average of sense embeddings in } R_{\psi'};$
            $\texttt{Insert}(\vec{\psi}', \Psi);$ `// add to represented`
            $\texttt{Remove}(\psi', \Psi');$ `// remove from unrepresented`

    $\textbf{return } \Psi, \Psi';$

$\Psi, \Psi' \leftarrow \textbf{Propagate } (\Psi, \Psi', \textit{\{all } (\psi, \psi') : \texttt{Synset}(\psi) = \texttt{Synset}(\psi')\})$
$\Psi, \Psi' \leftarrow \textbf{Propagate } (\Psi, \Psi', \textit{\{all } (\psi, \psi') : \texttt{Hypernym}(\psi) = \texttt{Hypernym}(\psi')\})$
$\Psi, \Psi' \leftarrow \textbf{Propagate } (\Psi, \Psi', \textit{\{all } (\psi, \psi') : \texttt{Lexname}(\psi) = \texttt{Lexname}(\psi')\})$

Since this method is designed to achieve full-representation of the sense inventory based on a subset of senses observed in context, the inferred representations are also of a similar contextual nature. However, unless the initial set of sense embeddings is nearly complete, and particularly diversified, this propagation method produces some number of identical representations for distinct senses, which is most undesirable for disambiguation applications, and to a lesser extent, sense matching applications as well.

4.2.2. Leveraging glosses and lemmas

In Loureiro & Jorge [61] we introduced a method for representing sense embeddings based on glosses and lemmas. This method is inspired by a typical baseline approach used in works pertaining to sentence embeddings, and it amounts to simply averaging the contextual embeddings for all tokens present in a sentence. In our case, we use glosses as sentences, but also introduce lemmas into the gloss' context. By combining glosses with lemmas, we not only augment the information available to represent senses, but we are also able to generate sense embeddings which are lemma-specific (sensekey-level), instead of only concept-specific (synset-level) if we only used glosses. As such, sense embeddings generated by this method address the redundancy issue arising from the previously described propagation method, while simultaneously introducing representational information which is complementary to contextual embeddings extracted from sense-annotated sentences.

The method proceeds as follows. For every lemma/sense pair (i.e., sensekey) in a sense inventory, we build the template "*<lemma>, <sense lemmas> - <sense gloss>*". For instance, based on WordNet, the synset $race_2^v$ has the lemmas *race* and *run* which are provided with the following sensekey-specific fill-outs of the template:

- *race%2:33:00:: -* "race - run, race - compete in a race"
- *run%2:33:01:: -* "run - run, race - compete in a race"

The initial "*<lemma>*" component of the template can be omitted if the target representation level is synsets, as it only serves the purpose of reinforcing the lemma which is specific to the sensekey. The templated string is processed by Ω, similarly to sentences S in Section 4.1, but here we use the resulting set of contextual embeddings for every token $C_\star$.

Considering that we have a complete set of sense embedding Ψ, based on sense annotations and propagation as described in Sections 4.1 and 4.2.1, we augment $\forall \vec{\psi} \in \Psi$ with gloss and lemma information as follows:

$$\vec{\psi} = \frac{1}{2}(||\vec{\psi}||_2 + ||\frac{1}{|C_\star|} \sum_{l \in L} \sum_{\vec{c} \in C_\star} \vec{c}_l||_2), \, where \, C_\star = \Omega(\text{Template}(\psi)) \tag{3}$$

In contrast to Loureiro & Jorge [61], which proposed using concatenation to merge this new set of sense embeddings based on glosses and lemmas with the previously mentioned set, in this work we propose merging through averaging instead. This departure is motivated by the fact that Loureiro & Jorge [61] found that while concatenation outperformed averaging for WSD, the difference in performance was modest, and in this work we are interested in additional tasks which that work did not cover. Merging representations through concatenation doubles the dimensionality of sense embeddings, increasing computational requirements and complicating comparison with contextual embeddings, among other potential applications. On the other hand, merging representations through averaging allows for adding more components while retaining a similar vector, of equal dimensionality to contextual embeddings, and represented in the same vector space.

4.3. Applying sense embeddings

In this section we address how sense embeddings can be employed for solving various tasks, grouped under two paradigms: disambiguation and matching. Disambiguation assigns a word in context (i.e., in a sentence) to a particular sense out of a subset of candidate senses, restricted by the word's lemma and part-of-speech. Matching also assigns specific senses to words, but imposes no restrictions, admitting every entry in the sense inventory for each assignment.

The different conditions for disambiguation and matching require sense representations with different degrees of lexical information and semantic coherence. Whereas, for disambiguation, lexical information can be absent from sense representations, due to the subset restrictions, for matching, lexical information is essential to distinguish between word forms carrying identical or similar semantics. Similarly, the disambiguation setting has no issues with sense representations displaying inconsistencies such as *eat* being more similar to *sleep* than to *drink*, since these all belong to disjoint subsets, but the order and coherence of these similarities is relevant for sense matching applications. This distinction leads us to specialize sense embeddings accordingly in Section 4.4.

To disambiguate a word w in context, we start by creating a set Ψ_w of candidate senses based on its lemma and part-of-speech, using information provided with the sense inventory. Afterwards, we compute the cosine similarities (denoted 'cos') between the word's contextual embedding $\vec{c}_w$ and the pre-computed embeddings for each sense in this subset Ψ_w (both using the same layer pooling). Finally, we assign the sense whose similarity is highest (i.e., nearest neighbor):

$$\Psi_w = \{\vec{\psi} \in \Psi : \text{lemma and part-of-speech of } \psi \text{ match } w\}$$

$$\text{Disambiguation}(w) = \underset{\vec{\psi} \in \Psi_w}{\arg\max}(\cos(\vec{c}_w, \vec{\psi})) \tag{4}$$

To match a word w in context, without restrictions, we follow the approach for disambiguation but simply consider the full sense inventory Ψ instead of Ψ_w:

$$\text{Matching}(w) = \underset{\vec{\psi} \in \Psi}{\arg\max}(\cos(\vec{c}_w, \vec{\psi})) \tag{5}$$

4.4. Grounding layer pooling

Up until this point, we have described our method closely following our prior work in Loureiro & Jorge [61]. As we covered in earlier sections, NLMs can show substantial and, more importantly, unexpected variation in task performance across their layers. Considering this work's focus on more principled and grounded sense representation with NLMs, our methodology also covers this important aspect.

In this section we present two methods targeting the layers composing NLMs. The first method probes each layer's adeptness for sense representation. Consequently, the second method in this section is designed to capitalize on that knowledge towards sense representations which better capture each NLM's ability to represent senses over the current paradigm.

As we alluded to in Section 4.3, sense representation should be viewed in light of the intended applications for these representations. In particular, in this work we differentiate between representations used for disambiguating words, and for matching or comparing senses. This distinction is motivated by the fact that disambiguation, which is the prevalent sense-related task on NLP, only requires that sense representations be adequately differentiated between the restricted set of senses which share the same lemmas and parts-of-speech. However, there exist other potential applications where sense representations are matched without any constraints on the sense inventory, and thus require that senses be coherently represented across the semantic space.

4.4.1. Sense probing

In order to assess the contribution of individual layers of a pre-trained NLM for sense representation, we directly evaluate the performance of representations from these layers on tasks related to the previously described disambiguation or matching scenarios. These tasks are solved using the nearest neighbors approaches described in Section 4.3, comparing pre-computed sense embeddings with contextual embeddings obtained from the same layer.

For this probing experiment, we follow the method for learning sense representations described in Section 4.1, but create multiple sets of senses Ψ_l for each layer l in the NLM. To maintain focus on assessing the performance of representations learned directly from specific layers, we ensure that test instances have all their senses represented in the sense-annotated corpora used to precompute $\Psi_l, \forall l \in L$. Thus, our probing experiments do not use techniques to infer or enrich sense representations, such as those we described in Section 4.3, which could otherwise act as confounders.

The resulting performance scores for every layer $l \in L$ composing a specific Ω, using a corresponding Ψ_l, not only reveal which layers perform best, but also inform the layer pooling method described next.

4.4.2. Sense profiling

We use the probing results described earlier as the basis for a pooling operation which is better grounded than the current paradigm of using the sum of the last four layers, and also better performing as we show later in this work. We designate each set of model-specific layer weights as a 'sense profile', and consider distinct sense profiles for disambiguation and matching, depending on the choice of disambiguation or matching tasks during layer probing.

These proposed sense profiles are a more immediate version of the Scalar Mixing used in Tenney et al. [111], being based on heuristically-derived sets of layer weights, instead of learning them through task optimization. Considering this, we understand sense profiles to be closer to the extraction configurations of Vulić et al. [119].

Granted we have performance scores $s_l \forall l \in L$, for a specific Ω, we obtain layer specific weights $w_l \forall l \in L$ by applying the softmax function:

$$w_l = \frac{\exp(s_l/t)}{\sum_{l' \in L} \exp(s_{l'}/t)} \tag{4}$$

We use the temperature scaling parameter t to skew the weight distribution towards highest performing layers. While simple, temperature scaling has been found surprisingly effective at calibrating neural network predictions [39]. This parameter is to be determined empirically and is only specific to application settings, not models.

In Table 1 we demonstrate the interaction between performance scores and layer weights conditioned on the temperature parameter. In that table, and others found in this work, we use reverse layer indices so that we can consistently refer to the final layer of any model using the -1 index, regardless of the number of layers in the NLM.

Consequently, we employ sense profiles comprised of weights $w_l \forall l \in L$ to retrieve contextual embeddings from Ω, and generate our sense embeddings accordingly, updating formula (1) such that:

$$\vec{\psi} = \frac{1}{|C_\psi|} \sum_{l \in L} \sum_{\vec{c} \in C_\psi} w_l * \vec{c}_l \, , \, where \, C_\psi = \Omega(S_\psi) \tag{5}$$

This set of sense embeddings learned from annotations using sense profiles, undergoes the same extensions and augmentations described earlier (§4.2).

To be clear, the process of probing layer performance and determining sense profiles to pool contextual embeddings from all layers (including when learning sense embeddings from annotations) is carried out for both the disambiguation and matching settings independently. As a result, we produce two sets of sense embeddings for each NLM based on sense profiles, which we distinguish from the LMMS sense embeddings introduced in Loureiro & Jorge [61] as LMMS-SP (Sense

Table 1

Shows interaction between F1 scores (rounded) for 1NN WSD using each layer of four different NLMs, and respective weight distributions (matching colors) using decreasing temperature parameters. Lower temperatures induce higher skewness towards layers that perform best on the probing validation set. Distributions based on t=1.000 are almost uniform, while t<0.002 would place almost all mass on single best layer. (For interpretation of the colors in the figure(s)/table(s), the reader is referred to the web version of this article.)

	INIT	-24	-23	-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
BERT-L	53	58	62	63	65	67	68	68	69	70	71	71	71	71	72	72	71	72	73	72	73	74	**75**	**75**	72
XLNet-L	51	57	65	67	68	70	71	72	**73**	73	72	72	72	72	71	71	71	71	71	71	72	72	72	72	68
RoBERTa-L	53	57	63	66	67	69	71	72	73	73	74	74	74	74	**75**	75	75	74	**75**	74	74	74	73	74	71
ALBERT-XL	54	65	67	68	69	70	70	**71**	**71**	**71**	**71**	**71**	**71**	**71**	**71**	70	70	69	69	69	69	69	69	69	64
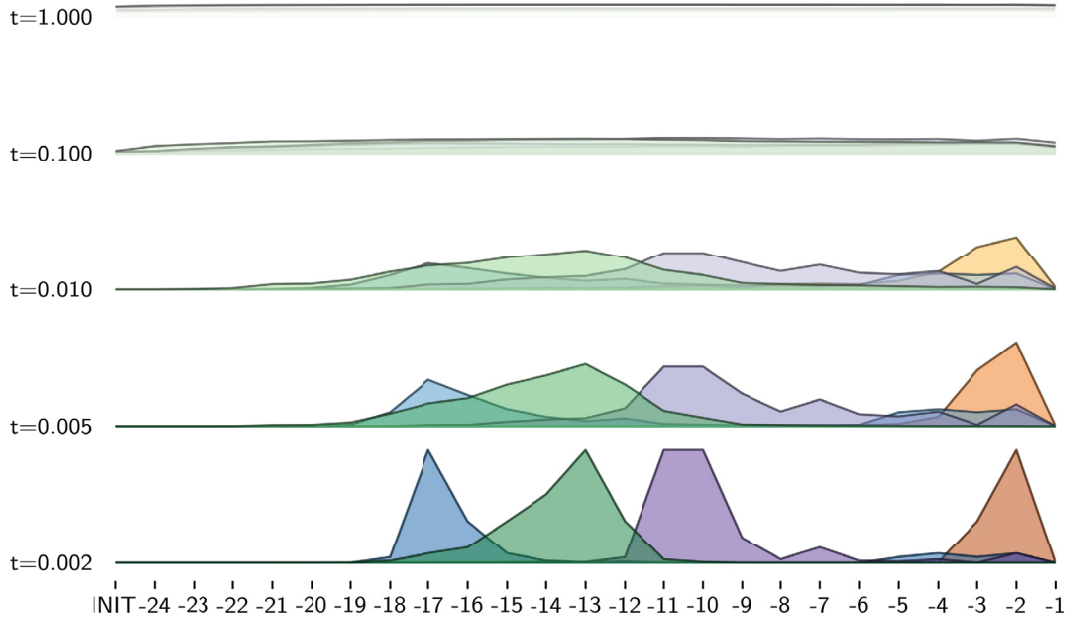
Profiles). The SP-WSD (for Word Sense Disambiguation) and SP-USM (for Uninformed Sense Matching) abbreviations are used to refer to sense embeddings based on disambiguation and matching sense profiles respectively.

5. Experimental setting

In this section we provide details about our experimental setting, including a description of the models (§5.1) and datasets (§5.2) used for learning sense representations.

5.1. Transformer-based language models

In this work we experiment with several Transformer-based Language Models, including all the original English BERT models released by Devlin et al. [29] as well as several other BERT-inspired alternatives, namely XLNet [124], RoBERTa [59] and ALBERT [52]. This section briefly describes the most relevant features of each of these models for our use case. We summarize the differences between each variant of these models on Table 2.

BERT. The model released by Devlin et al. [29] is first prominent Transformer-based NLM designed for language understanding. It is pre-trained with two unsupervised modeling objectives, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), using English Wikipedia and BookCorpus [128]. It uses WordPiece tokenization, splitting words into different components at the character-level (i.e., subwords). BERT is available in several models differing not only on parameter size, but also tokenization and casing. The 'whole-word' models were released after publication, showing slightly improved benchmark performance when trained with whole words being masked instead of subwords resulting from WordPiece tokenization.

Table 2

Feature comparison for the NLMs used in this work. Configuration names are shortened for readability: B - Base; L - Large; XL - Extra Large; XXL - Extra Extra Large; UNC - Uncased; WHL - Whole-Word.

Model	Configuration	Params.	Layers	Heads	Dims.	Tokenization	Tasks	Corpus
	B	110M	12	12	768	WordPiece	MLM, NSP	16 GB
	B-UNC	110M	12	12	768	WordPiece, Unc.	MLM, NSP	16 GB
BERT	L	340M	24	16	1024	WordPiece	MLM, NSP	16 GB
	L-UNC	340M	24	16	1024	WordPiece, Unc.	MLM, NSP	16 GB
	L-WHL	340M	24	16	1024	WordPiece	MLM, NSP	16 GB
	L-UNC-WHL	340M	24	16	1024	WordPiece, Unc.	MLM, NSP	16 GB
XLNet	B	110M	12	12	768	SentencePiece	PLM	158 GB
	L	340M	24	16	1024	SentencePiece	PLM	158 GB
RoBERTa	B	125M	12	12	768	Byte-level BPE	MLM	160 GB
	L	355M	24	16	1024	Byte-level BPE	MLM	160 GB
	B	11M	12	12	768	SentencePiece	MLM, SOP	160 GB
ALBERT	L	17M	24	16	1024	SentencePiece	MLM, SOP	160 GB
	XL	58M	24	16	2048	SentencePiece	MLM, SOP	160 GB
	XXL	223M	12	64	4096	SentencePiece	MLM, SOP	160 GB

XLNet. Based on a Transfomer-XL [26] architecture, Yang et al. [124] release XLNet featuring Permutation Language Modeling (PLM) as the only pre-training objective. The motivation for PLM is that it does not rely on masked tokens, and thus makes pre-training closer to fine-tuning for downstream tasks. It is also trained on much larger corpora than BERT, adding a large volume of web text from various sources to the corpora used for BERT. Instead of using WordPiece for tokenization, XLNet uses SentencePiece [50], which is a very similar open-source version of WordPiece.

RoBERTa. The model proposed by Liu et al. [59] is explicitly designed as an optimized version of BERT. RoBERTa does not use the NSP pre-training objective after finding that it deteriorates performance in the reported experimental setting, performing only MLM during pre-training. It is also trained with some different choices of hyperparameters (e.g., larger batch sizes) that improve performance on downstream tasks. The models released with RoBERTa are also trained on larger corpora composed mostly of web text, similarly to XLNet. As for tokenization, RoBERTa opts for byte-level BPE, following Radford et al. [92], which makes retrieving embeddings for specific tokens more challenging (i.e., spacing must be explicitly encoded).

ALBERT. Aiming for a lighter architecture, Lan et al. [52] propose ALBERT as a more parameter-efficient version of BERT. In spite of changes introduced to improve efficiency (e.g., cross-layer parameter sharing), ALBERT is based on a similar architecture to BERT. Besides improving efficiency, ALBERT also improves performance on downstream tasks by replacing NSP with the more challenging Sentence Order Prediction (SOP) objective. ALBERT uses the same SentencePiece tokenization as XLNet, and it is trained on similar corpora. It is released in several configurations, showing benchmark performance comparable to BERT while using fewer parameters.

The full set of 14 model variants detailed on Table 2 are only used for layer-specific validation performance on WSD and USM tasks. For task evaluation and analyses, we proceed with the single best performing model configuration from each model family, according to results from the validation experiments.

We use the Transformers package [122] (v3.0.2) for experiments with BERT, XLNet and ALBERT, and the fairseq package [79] (v0.9.0) for experiments with RoBERTa.[4]

5.2. Corpora for training and validation

We learn the initial set of sense representations described in Section 4.1 using sense-annotated corpora, namely SemCor [74] and the Unambiguous Word Annotations corpus (UWA) [60].

SemCor is a sense-annotated version of the Brown Corpus that still remains the largest corpus with manual sense-annotations despite its age. It includes 226,695 annotations for 33,362 sensekeys (25,942 synsets), reaching a coverage of 16.1% of WordNet's sense inventory. We use the version released in Raganato et al. [94], which includes mappings updated to WordNet version 3.0.

UWA is our recently introduced corpus composed exclusively of annotations for unambiguous words from Wikipedia sentences. Since WordNet is mostly composed of unambiguous words, UWA not only allows for representing the majority of WordNet senses (56.7%, when combined with SemCor) from direct annotations, but also leads to improved sense representation for senses learned through propagation (as described in Section 4.2.1), due to network effects. UWA is released in several versions of different sizes, in this work we use the version with up to 10 examples per sense (denoted UWA10), which includes 867,252 annotations for 98,494 sensekeys (67,860 synsets).

---

[4] Initial experiments with RoBERTa showed slightly better results using fairseq.

In order to avoid interference with the standard test sets, we perform our layer analysis and probing using a custom validation set, based on the MASC[5] corpus [46], following Loureiro et al. [63]. Considering that our layer experiments are focused on intrinsic properties of NLMs, this custom version of the MASC corpus is restricted to only include annotations for senses that occur in SemCor. Any sentence annotated with senses not occurring in SemCor is discarded, leaving a total of 14,645 annotations. As such, our layer experiments use sense embeddings learned from SemCor and validated using this restricted version of MASC, without requiring strategies for inferring senses (e.g., ontological propagation), or fallbacks (e.g., Most Frequent Sense).

6. Probing analysis

In this section we present the outcome of the probing methodology described in Section 4.4 applied on the models detailed in Section 5.1. We report probing results in this section so they are presented and discussed before the evaluation and analysis sections, which report downstream task results using layer pooling informed by the probing analysis.

This section starts by covering our initial findings regarding layer performance variation patterns observed for all models (§6.1). Second, we present validation results using our proposed sense profiles for both disambiguation and matching scenarios (§6.2). Finally, we present our rationale for choosing which sense profile should be used according to the type of task (§6.3).

6.1. Variation in layer performance across NLMs

As discussed in Section 3.3, it is well-understood that task performance varies considerably depending on which layers are used to retrieve embeddings from. While some works have analyzed task performance per layer specifically for the task of WSD [95,63], there is still lacking an in-depth cross-model comparison.

In Table 3 we report WSD and USM performance for individual layers of each of the 14 models belonging to 4 different Transformer-based model families. These results are obtained using the methodology described in Section 4.4.1. We observe that indeed, the final layer (-1) is never optimal for either WSD or USM performance. More interestingly, we find that while the second-to-last layer (-2) always performs best for WSD in BERT models (with the exception of BERT-L-UNC-WHL), that pattern does not hold for the other models tested. For some models, such as XLNet-L or ALBERT-L, we even find that the best performing layers are closer to the initialization layer (INIT) than to the final layer. Another apparent pattern is that the best performing layers for USM are consistently lower than for WSD. This can be explained by the fact that USM benefits from lexical information encoded in the lower layers, even though the initialization layer still performs worst, just as with WSD, demonstrating that it is not sufficient by itself.

These empirical results suggest that any layer pooling strategy based on a fixed set of layers, such as the often used sum of layers [-1,-4], cannot accurately capture the available sense information encoded in pre-trained NLMs.

6.2. Sense profiles for disambiguation and matching

In Section 4.4.2 we described our method for uncovering a model-specific set of layer weights which informs a weighted layer pooling that results in improved sense representations. We have applied this method to all our models, for both disambiguation and matching scenarios, in order to verify whether our proposed method reliably improves performance on WSD and USM tasks when compared to conventional pooling approaches. Additionally, we also compare against different values for the temperature t parameter. In order to understand whether our recommended t values actually result in improved test-time performance for these tasks, we run a limited evaluation on the ALL test set of Raganato et al. [94] where we compare NLMs using only our method as described until Section 4.2.1 (without using glosses) and trained solely with SemCor annotations. Later, in Sections 7.1 and 7.2, we report WSD and USM results using our final solution in comparison with the current state-of-the-art.

The conventional layer choices we considered are the following: last/final (L_{-1}); second-to-last (L_{-2}); sum of last 4 ($L_{-1} + L_{-2} + L_{-3} + L_{-4}$); integer weighted sum of last 4 ($L_{-1} + 2 * L_{-2} + 3 * L_{-3} + 4 * L_{-4}$); fractional weighted sum of last 4 ($\frac{1}{4} * L_{-1} + \frac{1}{3} * L_{-2} + \frac{1}{2} * L_{-3} + L_{-4}$). We tested temperature values $t \in \{0.002, 0.005, 0.01, 0.1, 1.0\}$.[6] Below we discuss our findings regarding the impact of sense profiles specific to each task.

The WSD validation results on Table 4 reveal that the single best layer (which varies depending on model, see Table 3) consistently outperforms the sum of last 4 layers. We also find that WSD sense profiles with $t = 0.002$ and $t = 0.005$ perform comparably to the single best layer, with $t = 0.002$ being slightly closer on average. Given the close performance, we opt for recommending $t = 0.005$ as higher values are less likely to overfit on the validation set (bias-variance tradeoff). In the limited evaluation results on Table 5 we compare the performance of conventional layer choices against WSD sense profiles with the recommended temperature value. We observe that for 11 out of 14 models, WSD sense profiles with the recommended temperature reliably outperform any of the conventional choices, of which none stands out as a reliable

---

[5] We use the version of the MASC corpus released in Vial et al. [114].
[6] $t = 0.001$ results in large exponents that cause overflow errors.

Table 3

Performance variation on the development set across all models and configurations considered in this work. Green represents best performing layers (best is marked with ⋆), red represents worst performing layers, and grey stands for layers missing in shallower variants.

Word Sense Disambiguation (WSD)

Model		INIT	-24	-23	-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
BERT	B																								⋆	
	B-UNC																								⋆	
	L																								⋆	
	L-UNC																							⋆	⋆	
	L-WHL																								⋆	
	L-UNC-WHL																		⋆							
XLNet	B																							⋆		
	L									⋆																
RoBERTa	B																							⋆		
	L														⋆	⋆										
ALBERT	B																			⋆		⋆				
	L										⋆	⋆														
	XL												⋆													
	XXL																			⋆						

Uninformed Sense Matching (USM)

Model		INIT	-24	-23	-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
BERT	B																		⋆	⋆	⋆					
	B-UNC																		⋆							
	L													⋆												
	B-UNC													⋆												
	L-WHL													⋆												
	L-UNC-WHL													⋆												
XLNet	B																		⋆	⋆						
	L									⋆																
RoBERTa	B																		⋆							
	L								⋆																	
ALBERT	B																		⋆							
	L										⋆															
	XL										⋆															
	XXL																	⋆	⋆							

cross-model choice. Moreover, on Table 5 we also see that WSD sense profiles with the recommended temperature generally match or outperform both the single layers which performed best on the validation set, and the WSD sense profiles using the temperature value that showed best performance on the validation set.

Our findings regarding performance of USM sense profiles largely follow the previously mentioned findings for WSD sense profiles. In the case of USM, validation results on Table 6 more clearly show that $t = 0.100$ performs better, although $t = 1.000$ also performs well. As for the limited evaluation results, Table 7 shows that conventional layer choices significantly underperform any of the alternatives introduced in this work, with the USM sense profile with recommended temperature ($t = 0.100$) showing overall best performance.

6.3. Choosing sense profiles for different tasks

Having established that our proposed sense profiles improve WSD and USM performance over conventional layer choices, the question remains of whether to choose WSD or USM sense profiles to represent sense embeddings. In this work we propose choosing sense profiles based on the probing task that shares most similar constraints to the downstream task of interest. More specifically, tasks requiring comparison of different senses for the same word fit the disambiguation profile, such as classical WSD [75] or WiC [88], and benefit less from information in lower layers. On the other hand, tasks without lexical constraints, not only USM but also synset similarity [20] or semantic change [40], are better suited to the

Table 4

WSD validation results (F1). Reports best single layer and weighted sums using specific sense profiles, with different t values, for each model configuration. Sense representations for this experiment were learned from SemCor (no propagation required).

Model		Sum LST4	Layer Best	Weighted Sum (WS)				
				t=0.002	t=0.005	t=0.010	t=0.100	t=1.000
BERT	B	71.6	**72.5** (-2)	72.4	72.2	71.8	69.6	68.6
	B-UNC	71.9	**73.0** (-2)	72.9	72.8	72.3	70.3	69.1
	L	73.8	**74.7** (-2)	**74.7**	74.5	74.3	72.2	70.9
	L-UNC	72.7	**72.9** (-2)	**72.9**	72.7	72.7	71.5	70.8
	L-WHL	72.0	73.0 (-2)	**73.1**	72.6	71.8	70.4	69.1
	L-UNC-WHL	71.5	**72.7** (-8)	72.6	72.1	72.0	71.5	70.7
XLNet	B	66.6	70.9 (-3)	71.0	**71.2**	**71.2**	70.3	69.9
	L	66.5	72.7 (-17)	73.0	**73.4**	73.3	72.4	71.8
RoBERTa	B	72.5	**72.9** (-3)	72.8	72.6	72.2	71.6	71.2
	L	74.1	**74.9** (-10)	**74.9**	74.7	74.4	73.6	73.1
ALBERT	B	68.3	**68.9** (-5)	68.3	68.2	68.1	67.6	67.3
	L	69.4	**70.5** (-15)	70.2	70.0	69.9	69.3	69.3
	XL	68.2	**71.4** (-13)	71.1	71.1	71.0	70.5	70.4
	XXL	72.4	**73.8** (-6)	73.5	73.4	73.3	72.8	72.4

Table 5

WSD test results (F1 on ALL). Reports conventional layer choices and alternatives using sense profiles. Recommended t for USM is 0.005. Sense representations for this experiment were learned from SemCor (with propagation).

Model		Standard					Proposed		
		Layer -1	Layer -2	Sum LST4	WS (I) LST4	WS (F) LST4	Layer Best Dev	WS Rec. t	WS Best t
BERT	B	72.1	**72.9**	72.6	72.5	72.5	**72.9** (-2)	72.8	**72.9** (.002)
	B-UNC	**73.5**	**73.5**	73.0	73.3	73.3	**73.5** (-2)	73.4	**73.5** (.002)
	L	73.3	73.9	74.0	74.0	74.0	73.9 (-2)	**74.2**	74.0 (.002)
	L-UNC	73.4	73.6	73.9	**74.0**	**74.0**	73.6 (-2)	**74.0**	73.8 (.002)
	L-WHL	72.0	73.4	73.2	73.1	73.0	73.4 (-2)	**73.5**	73.4 (.002)
	L-UNC-WHL	72.0	73.0	72.9	72.8	72.8	65.4 (-8)	**73.1**	73.1 (.002)
XLNet	B	69.1	67.4	64.8	62.7	63.7	55.4 (-3)	**72.3**	72.3 (.005)
	L	66.2	70.4	65.7	64.8	66.1	57.5 (-17)	**73.8**	73.8 (.005)
RoBERTa	B	71.9	73.3	73.3	73.4	73.3	73.5 (-3)	73.6	**73.7** (.002)
	L	71.2	74.0	74.1	74.0	73.9	66.3 (-10)	**74.7**	74.7 (.002)
ALBERT	B	**70.6**	69.6	70.1	70.1	70.3	67.3 (-5)	69.7	69.7 (.002)
	L	70.1	70.5	70.5	70.6	70.4	67.7 (-15)	**71.1**	70.7 (.002)
	XL	64.3	69.0	68.8	67.8	67.2	66.6 (-12)	**73.0**	73.0 (.002)
	XXL	69.4	73.7	73.9	73.1	72.5	74.8 (-6)	**75.1**	75.1 (.002)

matching profile, which uses information from more layers. In Section 7 we evaluate sense embeddings learned using sense profiles according to each task's constraints, and in Section 8.1 we analyze the performance gap when using alternate sense profiles.

7. Evaluation

In this work we address several sense-related tasks selected to investigate the versatility of the proposed sense embeddings, covering disambiguation (§7.1 - WSD), matching (§7.2 - USM), meaning change detection (§7.3 and §7.4 - WiC and GWCS) and sense similarity (§7.5 - SID). For each task, we report our new results (LMMS-SP) in comparison with the state-of-the-art and the original LMMS [61] sense embeddings.

For brevity, we only consider the variant from each model family that showed best results in our probing analysis (§6). In our comparisons, we omit LMMS$_{2348}$ because those sense embeddings are concatenated with fastText [12] word embeddings, thus not exclusively based on representations from particular NLMs, as focused in this work.

All tasks are solved essentially using cosine similarity between contextual embeddings and LMMS-SP precomputed sense embeddings represented using the same NLM. Each task's subsection provides more details about how these similarities are used to produce task-specific predictions. No additional task-specific training or validation datasets are used asides from those referred in Section 5.2, and all NLMs are employed in the same exact fashion - simply retrieving contextualized representations from each layer (following §4.1).

Table 6

USM validation results (F1). Reports best single layer and weighted sums using specific sense profiles, with different *t* values, for each model configuration. Sense representations for this experiment were learned from SemCor (no propagation required).

Model		Sum LST4	Layer Best	Weighted Sum (WS)				
				t=0.002	t=0.005	t=0.010	t=0.100	t=1.000
BERT	B	59.2	61.2 (-6)	61.5	61.5	61.8	**62.2**	62.0
	B-UNC	58.6	63.1 (-8)	63.1	63.2	63.5	**63.7**	**63.7**
	L	57.7	63.8 (-14)	63.9	64.0	63.9	**64.5**	**64.5**
	L-UNC	56.8	64.3 (-14)	64.4	64.6	64.6	65.4	**65.7**
	L-WHL	59.7	62.1 (-14)	62.1	62.0	62.1	**62.7**	62.5
	L-UNC-WHL	59.1	**64.8** (-14)	64.7	64.5	64.5	64.7	**64.8**
XLNet	B	34.7	61.2 (-9)	61.3	61.4	61.4	**61.9**	60.4
	L	28.0	63.6 (-18)	63.5	63.6	63.7	**64.4**	64.1
RoBERTa	B	61.6	**64.2** (-9)	64.0	63.9	64.0	64.0	63.9
	L	64.1	65.3 (-17)	65.2	65.4	65.8	66.1	**66.2**
ALBERT	B	60.0	61.8 (-8)	**61.9**	61.8	61.7	61.6	61.6
	L	60.0	**64.1** (-16)	63.5	63.5	63.2	63.3	63.2
	XL	54.9	64.5 (-18)	64.2	64.1	64.2	64.5	**64.6**
	XXL	65.8	65.8 (-9)	65.7	66.1	66.1	66.2	**66.3**

Table 7

USM test results (F1 on ALL). Reports conventional layer choices and alternatives using sense profiles. Recommended *t* for USM is 0.1. Sense representations for this experiment were learned from SemCor (with propagation).

Model		Standard					Proposed		
		Layer -1	Layer -2	Sum LST4	WS (I) LST4	WS (F) LST4	Layer Best Dev	WS Rec. t	WS Best t
BERT	B	53.1	51.2	53.7	53.0	53.0	57.0 (-6)	**57.7**	**57.7** (.100)
	B-UNC	50.4	50.4	53.0	51.8	52.3	57.9 (-8)	**58.8**	**58.8** (.100)
	L	53.9	50.3	52.5	52.8	53.4	58.9 (-14)	**60.0**	**60.0** (.100)
	L-UNC	48.3	46.7	49.0	48.8	48.8	58.5 (-14)	60.3	**60.4** (1.00)
	L-WHL	53.3	54.3	54.5	54.2	54.5	57.6 (-14)	**58.4**	**58.4** (.100)
	L-UNC-WHL	53.7	52.4	53.3	53.3	53.3	58.3 (-14)	59.4	**59.6** (1.00)
XLNet	B	38.1	36.9	31.4	27.8	28.9	**57.3** (-9)	**57.3**	**57.3** (.100)
	L	27.9	41.3	28.7	28.2	29.6	59.0 (-18)	**60.4**	**60.4** (.100)
RoBERTa	B	53.7	53.2	55.1	55.3	55.5	58.3 (-9)	**59.2**	**59.2** (.100)
	L	56.3	56.7	57.9	58.1	58.0	60.6 (-17)	**61.2**	**61.2** (.100)
ALBERT	B	53.8	53.6	54.7	54.2	54.8	55.8 (-8)	56.2	**56.3** (.002)
	L	55.7	55.5	56.0	56.1	56.1	57.4 (-16)	**58.4**	57.3 (.005)
	XL	41.2	48.5	49.8	48.0	47.4	59.9 (-18)	**60.1**	59.8 (1.00)
	XXL	55.1	60.6	61.7	61.0	60.5	60.6 (-9)	**62.3**	62.3 (1.00)

Table 8

Example WSD instance from Raganato et al. [94]. Sentence, lemma and part-of-speech (POS) are provided. The goal is to predict the correct sensekey (sense from WordNet).

Sentence	Lemma	POS	Gold Sensekey
Eyes that were clear, but also bright with a strange intensity, a sort of cold **fire** burning behind them.	fire	NOUN	fire%1:12:00::

As such, LMMS-SP performance on these tasks should be indicative of each NLM's intrinsic ability to approximate meaning representations learned during pre-training with language modeling objectives alone.

7.1. Word sense disambiguation (WSD)

WSD is the most popular and obvious task for evaluating sense embeddings. This task has been researched since the early days of Artificial Intelligence and constitutes an AI-complete task [75]. It is usually formulated as choosing the correct sense for a word in context out of a list of possible senses given the word's lemma and part-of-speech tag (see Table 8). Several test sets have been proposed over the years, and the compilation of Raganato et al. [94] has emerged as the de facto evaluation framework for English WSD, which we also use. Naturally, this task suits sense profiles for WSD, and we follow the method described in Section 4.3.

Table 9
F1 scores (%) for each test set in the WSD Evaluation Framework [94]. Top rows show results for models using sense annotations exclusively from SemCor (SC). Bottom rows show results for models augmenting SC with annotations from additional sources. Results marked with * correspond to development sets (and therefore ALL). For each group of results, we underline the best from LMMS.

	Model	1NN	Defs.	Rels.	SE2 (n=2,282)	SE3 (n=1,850)	SE07 (n=445)	SE13 (n=1,644)	SE15 (n=1,022)	ALL (n=7,253)
SemCor (SC)	MFS				65.6	66.0	54.5	63.8	67.1	64.8
	context2vec [68]	✓			71.8	69.1	61.3	65.6	71.9	69.0
	ELMo [84]	✓			71.6	69.6	62.2	66.2	71.3	69.0
	BERT-L [61]	✓			76.3	73.2	66.2	71.7	74.1	73.5
	SVC [115]			✓	76.6	76.9	69.0	73.8	75.4	75.4
	GlossBERT [44]		✓		77.7	75.2	72.5*	76.1	80.4	77.0*
	EWISER [9]		✓	✓	78.9	78.4	71.0	78.9	79.3*	78.3*
	BEM [11]		✓		79.4	77.4	74.5*	79.7	81.7	79.0*
	ConSeC [5]		✓		82.3	79.9	77.4*	83.2	85.2	82.0*
	LMMS$_{1024}$ [61]	✓		✓	75.4	74.0	66.4	72.7	75.3	73.8
	LMMS$_{2048}$ [61]	✓	✓	✓	76.3	<u>75.6</u>	68.1	75.1	77.0	<u>75.4</u>
	LMMS-SP$_{BERT-L}$	✓	✓	✓	76.1	74.0	67.0	75.2	<u>77.4</u>	75.0
	LMMS-SP$_{XLNet-L}$	✓	✓	✓	76.0	73.1	66.4	74.2	74.9	74.1
	LMMS-SP$_{RoBERTa-L}$	✓	✓	✓	77.2	73.5	67.9	<u>75.5</u>	76.4	75.2
	LMMS-SP$_{ALBERT-XXL}$	✓	✓	✓	<u>77.4</u>	74.8	<u>71.0</u>	74.7	74.8	<u>75.4</u>
SC+Others	SVC [115]			✓	79.4	78.1	71.4	77.8	81.4	78.5
	KnowBERT [86]			✓	76.4	76.0	71.4	73.1	75.4	75.1
	EWISER [9]		✓	✓	80.8	79.0	75.2	80.7	81.8*	80.1*
	ARES [105]	✓	✓		78.0	77.1	71.0	77.3	83.2	77.9
SC+UWA	LMMS-SP$_{BERT-L}$	✓	✓	✓	76.7	74.1	66.4	75.2	<u>77.6</u>	75.2
	LMMS-SP$_{XLNet-L}$	✓	✓	✓	76.1	73.1	65.9	74.2	75.0	74.1
	LMMS-SP$_{RoBERTa-L}$	✓	✓	✓	77.4	73.5	67.7	<u>75.3</u>	76.7	75.2
	LMMS-SP$_{ALBERT-XXL}$	✓	✓	✓	<u>77.7</u>	<u>75.0</u>	<u>70.5</u>	74.7	74.9	<u>75.5</u>

7.1.1. Results

On Table 9 we report performance on the standard test sets of the WSD Evaluation Framework [94]. Given the breadth of recent WSD solutions, we make results more comparable by separating solutions using only SemCor annotations, and solutions augmenting SemCor with other sense-annotated datasets. In the case of LMMS-SP, we combine SemCor with the unambiguous annotations from UWA, which are easily retrieved from unlabeled corpora. We also report which solutions use glosses and relations, besides sense annotations, as well as which solutions are based on 1NN (first nearest neighbor) with precomputed sense embeddings represented in the space of NLMs.

When considering SemCor as the only source of annotations, LMMS and LMMS-SP remain the best solutions based on 1NN in NLM-space. Most notably, LMMS-SP$_{ALBERT-XXL}$ is able to match the performance of LMMS$_{2048}$ on the combination of test sets (ALL) without concatenating gloss embeddings.

As could be expected, task-specific classifiers show best results, particularly BEM [11] and ConSeC [5]. Generally, solutions fine-tuning NLMs, or combining them with other classifiers trained for the WSD task show improved performance over 1NN. Despite this, in Loureiro et al. [63] we have shown that 1NN solutions offer other advantages, such as better sample efficiency and less frequency biases, besides the versatility advocated in this current work.

Allowing for additional annotations, we find that LMMS-SP results improve slightly when using BERT-L and ALBERT-XXL. ARES [105] uses semi-supervised annotations to increase coverage of the sense inventory with sense embeddings represented on the space of BERT-L. Results show the ARES dataset leads to improved WSD performance in comparison to LMMS-SP on the reported test sets, particularly on SE13 and SE15.[7]

7.2. Uninformed sense matching (USM)

We introduced the USM task in Loureiro & Jorge [61] as a variation on WSD that can more accurately represent the extent to which NLMs can associate words or phrases to senses from the WordNet inventory. The crucial difference in relation to WSD is that in the USM task we do not use any supplemental information to restrict candidates in the sense inventory (compare examples in Table 8 and Table 10). Conveniently, this allows for USM to use the same test sets as WSD. As expected, we address USM using the sense profile of the same name, and follow the method described in Section 4.3. In this work we evaluate USM from both the sensekey and synset perspective, to provide a clearer account of the impact of lexical information on task performance.

---

[7] We expect the annotations in ARES to produce further performance gains for LMMS-SP but do not use this resource due to its large size (13x the annotations in SemCor+UWA10).

Table 10

Example USM instance adapted from the WSD Evaluation Framework [94]. The correct sensekey or synset must be predicted, in separate evaluations.

Sentence	Gold Sensekey	Gold Synset
Eyes that were clear, but also bright with a strange intensity, a sort of cold **fire** burning behind them.	fire%1:12:00::	06711159n ($fire_n^9$)

Table 11

USM results on the ALL test set of the WSD Evaluation Framework [94], at sense and synset-level. Results marked with † are obtained from synset embeddings converted from sensekey embeddings.

Model	Sensekeys			Synsets		
	F1	P@5	MRR	F1	P@5	MRR
ARES	61.4	84.7	71.8	**60.7**†	**86.5**†	**71.8**†
LMMS$_{1024}$ [61]	52.2	66.9	59.0	29.4†	53.9†	40.7†
LMMS$_{2048}$ [61]	34.8	60.3	46.3	32.5†	58.9†	44.5†
LMMS-SP$_{BERT-L}$	60.8	86.7	72.2	51.0	81.7	64.3
LMMS-SP$_{XLNet-L}$	60.1	87.3	71.9	51.7	82.7	65.1
LMMS-SP$_{RoBERTa-L}$	62.2	86.9	73.1	50.2	80.1	63.3
LMMS-SP$_{ALBERT-XXL}$	**62.9**	**87.6**	**73.7**	52.7	81.9	65.5

Table 12

Examples from the WiC training set. Showing two independent instances.

Sentence Pairs	Lemma	POS	Boolean
You must **carry** your camping gear. Sound **carries** well over water.	carry	VERB	False
He wore a jock strap with a metal **cup**. Bees filled the waxen **cups** with honey.	cup	NOUN	True

7.2.1. Results

Following Loureiro & Camacho-Collados [60], we evaluate performance considering two additional metrics besides F1: Precision at 5 (P@5) and Mean Reciprocal Rank (MRR). To our knowledge, ARES [105] is the only other publicly available set of full-coverage sense embeddings represented in the space of a Transformer-based NLM, so we also compare LMMS-SP against those sense embeddings. Since our prior LMMS sense embeddings and the ARES sense embeddings are released using sensekey representations, USM synset evaluation requires converting those sensekey embeddings to synset embeddings. We perform this conversion by simply averaging sensekey embeddings that belong to the same synset. In Section 8.2 we analyze the impact this conversion can have on task performance.

On Table 11 it can be observed that LMMS-SP dramatically improves performance over LMMS on all three metrics considered. The poor performance of LMMS$_{2048}$ in comparison to LMMS$_{1024}$ suggests that concatenating gloss embeddings is detrimental to USM performance, particularly on the F1 metric. In this comparison we do not consider LMMS$_{2348}$ because those sense embeddings are concatenated with fastText static embeddings, resulting in 300 dimensions having the same exact distribution for sense embeddings corresponding to identical lemmas. This property of LMMS$_{2348}$ makes the comparison inequitable and diverts from this work's focus on the intrinsic capabilities of Transformer NLMs.

Interestingly, we find that, when targeting sensekeys, LMMS-SP$_{ALBERT-XXL}$ shows best performance on all metrics, and ARES (based on BERT-L) only outperforms LMMS-SP$_{BERT-L}$ on the F1 metric. However, when targeting synsets, the additional contexts of ARES prove more advantageous, and we do not observe a similar performance gap between sensekeys and synset as we do with LMMS-SP, which can be expected considering that the additional contexts of ARES are targeted at the synset-level.

7.3. Word-in-context (WiC)

The Word-in-Context [88, WiC] task is designed to assess how context impacts word representations produced by contextual NLMs. It is a binary classification task that simply requires determining whether a particular word is used with the same meaning or not in a pair of sentences, also given lemma and POS provided in WSD tasks (see Table 12 for examples). The dataset is balanced and performance is measured with accuracy.

7.3.1. Solution

In this work, we tackle the WiC task using our proposed sense embeddings following the unsupervised approach from Loureiro & Jorge [62], which essentially applies the 1NN method for disambiguating the target word in both sentences and checks whether they are equal or not. Even though we also explored a supervised approach in Loureiro & Jorge [62], based on Logistic Regression, in this work we focus on the unsupervised approach as its performance is more revealing of the inherent representational abilities of NLMs. Given the close relation to disambiguation, we use WSD sense profiles for WiC.

Table 13

Results (Accuracy) on the test set of the WiC task comparing our unsupervised approach to the state-of-art. Best results for each approach reported in bold. Our results were obtained from the Codalab online platform. Results marked with * used the SuperGLUE version of the WiC test set, which has minor preprocessing differences.

	Method	Language Model	Sense Embeddings	Acc.
Supervised	Fine-Tuning	BERT-L [120]	-	69.6*
	Logistic Reg.	BERT-L	LMMS$_{2048}$ [62]	68.1
	Fine-Tuning	RoBERTa-L [59]	-	69.9*
	Fine-Tuning	KnowBERT [86]	-	70.9
	Fine-Tuning	SenseBERT [54]	-	72.1
	Fine-Tuning	T5 [93]	-	**76.9***
	Fine-Tuning	BERT-L	ARES [105]	72.2*
Threshold	1NN WSD	-	JBT [81]	53.6
	1NN WSD	-	DeConf [90]	58.7
	1NN WSD	context2vec [68]	-	**59.3**
	1NN WSD	-	SW2V [65]	58.1
	1NN WSD	ELMo [84]	-	57.7
	1NN WSD	-	LessLex [21]	59.2
Unsupervised	1NN WSD	BERT-L	ARES (2020)	67.6
	1NN WSD	BERT-L	LMMS$_{2048}$ [61]	66.3
	1NN WSD	BERT-L	LMMS-SP$_{BERT-L}$	67.4
	1NN WSD	XLNet-L	LMMS-SP$_{XLNet-L}$	66.1
	1NN WSD	RoBERTa-L	LMMS-SP$_{RoBERTa-L}$	67.8
	1NN WSD	ALBERT-XXL	LMMS-SP$_{ALBERT-XXL}$	**67.9**

7.3.2. Results

WiC is a benchmark NLU task, being part of SuperGLUE [120], therefore most state-of-the-art NLMs have reported results for this task. The initial baseline methods proposed with WiC were based on cosine similarity with thresholds learned from the validation set.

Most recent solutions, however, involve fine-tuning the NLM (as performed for other sentence classification tasks in SuperGLUE) using the training and validation sets provided with WiC. One notable exception is Scarlini et al. [105] which proposed a method that leverages ARES sense embeddings to improve the fine-tuning process. As such, on Table 13 we compare results from these solutions to our unsupervised LMMS and LMMS-SP, as well as an unsupervised result based on the same 1NN approach using the ARES embeddings.

Starting with our unsupervised results, we confirm that LMMS-SP$_{BERT-L}$ surpasses the performance of LMMS$_{2048}$ (based on BERT-L), and once again LMMS-SP$_{ALBERT-XXL}$ displays the best performance. Nevertheless, supervised solutions using NLMs fine-tuned for this task show best performance overall, particularly T5 [93] which is currently the largest NLM with reported results on this task, at over 11B parameters. KnowBERT [86] and SenseBERT [54] are both NLMs based on BERT that have been augmented with sense information from WordNet and SemCor, among other resources, showing improved performance in comparison to fine-tuning the original BERT-L.

The method used by Scarlini et al. [105] to employ sense embeddings while fine-tuning BERT-L for WiC resulted in a notable improvement similar to SenseBERT. In the unsupervised setting, however, we found that ARES embeddings outperform LMMS-SP$_{BERT-L}$, but underperform both LMMS-SP$_{RoBERTa-L}$ and LMMS-SP$_{ALBERT-XXL}$. We expect following the same method to assist supervised fine-tuning with LMMS-SP sense embeddings may produce improved results, but consider that experiment out of scope for this work.

As for solutions using the threshold method, all reported models substantially underperform unsupervised results using any Transformer-based NLM.

7.4. Graded word similarity in context (GWCS)

For evaluating graded contextual similarity, in contrast to the binary contextual similarity assignments of WiC, we address SemEval 2020 Task 3: Graded Word Similarity in Context (GWCS) [2]. This task, based on the CoSimLex resource [3], targets word pairs used for evaluating distributional semantic models (not necessarily polysemous words) in contexts spanning multiple sentences. The task is divided into two sub-tasks derived from human-annotated similarity ratings: 1) predict the change in similarity between two different contexts for each word pair; 2) predict the similarity ratings themselves. Table 14 shows a single example from GWCS, featuring two contexts each with occurrences of the same pair of words, context specific similarity ratings, and the associated similarity change.

7.4.1. Solution

While the sub-tasks are independently evaluated, we employ essentially the same method for both, based on our straightforward approach for the WiC task covered in Section 7.3, with minor adjustments to quantify the observed change in similarity. Given contexts A and B, we disambiguate target words (each instance's word pair) in the corresponding contexts, and compute sense similarities sim_{wsd}^{A} and sim_{wsd}^{B} as the cosine similarity between the embeddings of the predicted

Table 14

Example from the practice set of GWCS (single instance). Contexts A and B each have corresponding similarity ratings for the same 'keep'-'protect' word pair.

	Contexts	Sim.	Change
A	Tim Drake **keeps** a memorial for her in his cave hideout underneath Titans Tower in San Francisco. [...] It is later revealed that Dr. Leslie Thompkins had faked her death after the gang war in an effort to **protect** her.	4.44	-0.52
B	Shisa are wards, believed to **protect** from various evils. When found in pairs, the shisa on the left traditionally [...] The open mouth to ward off evil spirits, and the closed mouth to **keep** good spirits in.	3.92	

Table 15

Results on both subtasks of SemEval 2020 Task 3. [†] as of 03/2021, considering evaluation and post-evaluation submissions (Users: [1]Ferryman, [2]Alexa). ARES results obtained using the same method as LMMS, only replacing the corresponding sense embeddings.

Model	Subtask1	Subtask2
Leaderboard Best[†]	77.4[1]	74.6[2]
ARES [105]	76.9	74.5
LMMS$_{1024}$ [61]	74.1	74.2
LMMS$_{2048}$ [61]	75.7	74.5
LMMS-SP$_{BERT-L}$	76.2	74.4
LMMS-SP$_{XLNet-L}$	**78.7**	**76.6**
LMMS-SP$_{RoBERTa-L}$	75.7	74.9
LMMS-SP$_{ALBERT-XXL}$	75.2	71.8

senses. Considering that disambiguation may predict the same senses, thus resulting in $sim_{wsd}^A = sim_{wsd}^B$ for many instances, we also compute contextual similarities sim_{ctx}^A and sim_{ctx}^B as the cosine similarity between the contextual embeddings of the target words. Thus, we determine similarity scores specific to context A as $sim^A = \frac{1}{2}(sim_{wsd}^A + sim_{ctx}^A)$, and similarity scores specific to context B as $sim^B = \frac{1}{2}(sim_{wsd}^B + sim_{ctx}^B)$. These context-specific similarities constitute our solutions to sub-task 2. We determine the semantic change scores for sub-task 1 trivially as $sim^B - sim^A$. Considering that this solution closely follows our solution for WiC, and that word pairs contained in this dataset tend to be closely related, we use the WSD sense profile for GWCS.

7.4.2. Results

Performance on sub-task 1 is measured with Pearson Uncentered Correlation between the system's scores and the average human annotations, and performance on sub-task 2 is measured with the harmonic mean of the Spearman and Pearson correlations between the system's scores and the average human annotations. On Table 15 we report results using our sense embeddings (LMMS and LMMS-SP), using the ARES sense embeddings with our scoring method (and BERT-Large, pooling with the sum of last 4 layers), and the best reported results from other task participants (including post-evaluation, until 03/2021). Similarly to WiC, the scores for the test sets are hidden from participants, both during evaluation (ended 03/2020) and post-evaluation periods (extends indefinitely), so all reported results are obtained from the online platform used by SemEval after submitting each system's predictions.

We observe that our straightforward method combining similarity between sense and contextual embeddings is able to outperform the solutions of other task participants (Leaderboard Best), most of which also relied on Transformer-based NLMs [2]. Interestingly, GWCS shows wide variation in performance from the choice of NLM, with LMMS-SP$_{XLNet-L}$ standing out with clearly best results on both sub-tasks.[8]

7.5. Sense similarity

All the tasks we considered so far (WSD, USM, WiC and GWCS) have evaluated sense embeddings by their utility for accurately matching or distinguishing word senses in particular contexts. In this last task, we address intrinsic evaluation of sense embeddings, directly comparing distributional similarity between sense pairs against human similarity ratings.

We perform this evaluation using the Sense Identification Dataset [22, SID], which is based on the word pairs (nouns only) and human similarity ratings from SemEval-2017 Task 2 [16], with the addition of mapping word pairs to particular senses in the BabelNet sense inventory (see examples on Table 16).

---

[8] Complete leaderboard results are available on Appendix A. Additionally, Appendix B reports performance on the Stanford Contextual Word Similarities [43] task, which inspired the GWCS and WiC tasks, with similar conclusions.

Table 16

Two examples of paired synsets with human similarity ratings from the SID dataset. Showing synset identifiers after conversion to WordNet (more readable format in parenthesis).

Synset 1	Synset 2	Similarity
08570634n (hayfield$_n^1$)	08598301n (grassland$_n^1$)	3.58
03169390n (decoration$_n^1$)	03291741n (envelope$_n^2$)	0.08

Table 17

Performance (Pearson Correlation) on the adapted SID dataset. All reported embeddings feature 300 dimensions. Embeddings marked with † have been converted from sensekeys. LessLex and NASARI embeddings were converted from BabelNet to WordNet using the same mapping applied to the SID adaptation.

	Synset Embeddings	WN Full Coverage	All (n=377)	Overlapping		
				All (n=354)	Polarized (n=182)	Observed (n=297)
Static	fastText [12]	✓	64.4	63.5	69.3	65.4
	NASARI$_{UMBC}$ [17]		-	71.6	79.1	74.4
	DeConf† [90]		75.1	74.9	80.6	76.9
	LessLex [21]		**82.5**	82.3	85.5	85.1
Contextual	SensEmBERT [104]		66.9	66.8	74.6	69.5
	ARES†	✓	70.6	70.4	80.5	73.3
	LMMS$_{2048}$† [61]	✓	71.2	72.2	76.2	76.3
	LMMS-SP$_{BERT-L}$	✓	77.8	77.8	80.4	83.1
	LMMS-SP$_{XLNet-L}$	✓	**79.5**	**79.6**	**81.2**	**84.5**
	LMMS-SP$_{RoBERTa-L}$	✓	74.1	74.2	79.0	80.9
	LMMS-SP$_{ALBERT-XXL}$	✓	77.4	77.2	80.5	81.4

7.5.1. Task adaptation

We convert the BabelNet sense identifiers to synsets from WordNet 3.0 using the mapping provided by Navigli & Ponzetto [76]. However, some instances cannot be mapped due to missing entries in WordNet, or, in rare cases, their mapping results in the two senses of the pair being equal, leading to a reduction of 492 instances to 377 mapped to Word-Net. We further split SID into different groups for additional insights. We first separate the 354 pairs for which both senses are represented in the related works we compare against (overlapping), considering these are not always complete sets of WordNet sense embeddings. Next, we breakdown the overlapping pairs into a set of the most polarized word pairs (i.e., similarity ratings ≤ 1 or ≥ 3), and another set containing only pairs with senses that are annotated in SemCor+UWA10 (observed).

7.5.2. Solution

We use cosine similarity between synset embeddings to correlate with human similarity ratings. Since we are directly comparing embeddings of very different dimensionality, we apply truncated SVD to normalize them to 300 dimensions[9] (including related work). The senses being compared range from completely unrelated (e.g., polyhedron$_n^1$; actor$_n^1$) to highly related or similar (e.g., actor$_n^1$; actress$_n^1$), so we use USM sense profiles for SID.

7.5.3. Results

Performance on SID is measured with Pearson correlation. For completeness, we report performance of synset embeddings that are not based on contextual NLMs, including new results based on fastText embeddings (trained on Common-Crawl).[10] Results for related works are based on sense embeddings provided by the authors, converting sensekeys to synsets by averaging the corresponding embeddings whenever required (as in Section 7.2). The inter-annotator agreement on our full set (n=377) reaches 87.9, measured as averaged pairwise Pearson correlation of the original SemEval-2017 human similarity scores.

Results for the WordNet-subset of SID are shown on Table 17. As can be observed, LMMS-SP substantially outperforms LMMS and related works. As with GWCS, LMMS-SP$_{XLNet-L}$ stands out with clearly best results. While LMMS-SP also outperforms most non-contextual embeddings, it still underperforms LessLex [21] embeddings, which are based on ensembles and learned using BabelNet. We also note that LMMS-SP performs particularly well on the 'Observed' set corresponding to senses learned from annotated corpora. The performance gap between ARES and LMMS-SP$_{BERT-L}$ suggests that additional semi-supervised annotations for more senses may not suffice. Finally, the 'Polarized' set seems consistently easier than the full set, indicating that the most challenging pairs are those with moderate similarity ratings.

---

[9] We verified that SVD-reduced embeddings always outperform original embeddings.

[10] fastText embeddings for a given synset are computed by averaging the word embeddings for each lemma that belongs to the input synset in WordNet.

Table 18

Impact of pooling operation on task performance. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

Task (Metric)	Pooling	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
WSD (F1 on ALL)	Sum-LST4	<u>75.2</u>	56.4	74.8	73.8
	SP-WSD ⋆	<u>75.2</u>	<u>74.1</u>	<u>75.2</u>	**<u>75.5</u>**
	SP-USM	72.9	73.4	74.2	74.5
USM (P@5 on ALL)	Sum-LST4	74.6	65.9	74.6	74.3
	SP-WSD	73.6	81.6	83.1	85.7
	SP-USM ⋆	<u>86.7</u>	<u>87.3</u>	<u>86.9</u>	**<u>87.6</u>**
WiC (ACC on Val.)	Sum-LST4	<u>71.8</u>	61.0	**<u>72.1</u>**	66.8
	SP-WSD ⋆	<u>71.8</u>	<u>67.9</u>	68.8	<u>68.7</u>
	SP-USM	67.7	65.0	68.5	67.9
GWCS (COR on ST2)	Sum-LST4	74.4	54.9	73.3	71.5
	SP-WSD ⋆	<u>76.3</u>	**<u>78.7</u>**	75.7	75.2
	SP-USM	73.4	75.4	<u>77.2</u>	<u>75.9</u>
SID (COR on ALL)	Sum-LST4	77.4	41.1	73.8	72.8
	SP-WSD	76.3	77.7	73.5	75.3
	SP-USM ⋆	<u>77.8</u>	**<u>79.5</u>**	<u>74.1</u>	<u>77.4</u>

8. Analysis

In this section, we perform several ablation studies to better understand the impact of individual contributions we have introduced in this work's extension of LMMS. These experiments target the same NLMs and tasks[11] that we addressed in the previous evaluation section. Our ablation analyses cover the impact of sense profiles (§8.1), UWA annotations (§8.2), merging gloss representations (§8.3) and indirect representation of synsets (§8.4). Considering that part-of-speech is an important factor in disambiguation (and sense representation), we also report performance per part-of-speech using both LMMS and LMMS-SP sense embeddings on WSD and USM tasks (§8.5).

8.1. Choice of sense profiles

On Table 18 we report performance according to the sense profile used for weighted pooling of contextual embeddings from NLMs (described in Section 6.2), and using the sum of the last 4 layers (Sum-LST4), as commonly used in related work and the original LMMS [61].

Our first conclusion is that Sum-LST4 pooling is only appropriate for particular models and tasks (i.e., WSD and WiC w/BERT-L; WiC w/RoBERTa-L), but detrimental for most (specially any task w/XLNet-L; any model for USM). However, our recommended choice of sense profile not only appears beneficial for WSD and USM tasks across all models (expected since the sense profiles are based on those tasks), but also for WiC, GWCS and SID. In fact, out of 20 model-task combinations, we only find 3 exceptions: RoBERTa-L on WiC and GWCS, and ALBERT-XXL on GWCS (to a lesser extent). Moreover, we confirm that tasks are sensitive to the choice between SP-WSD and SP-USM, which validate our task-specific recommendations.

8.2. Unambiguous word annotations

In this work we learnt our initial set of sense embeddings (as described in Sections 4.1 and 4.2.1) using SemCor, the only source of sense annotations used for LMMS [61], in combination with UWA [60], a set of sense annotations exclusively targeting unambiguous words.

On Table 19 we present results showing the impact of UWA on task performance. As noted in Loureiro & Camacho-Collados [60], the increase in WordNet coverage using UWA allows for disentangling dense clusters that coarsen the semantic space when relying on SemCor and network propagation alone. Consequently, we expect UWA to benefit sense matching tasks, which is confirmed by our results showing substantial improvements in USM and SID (the two tasks using SP-USM). We also find that UWA does not hinder performance on the remaining tasks for most model-task combinations (improves in most cases), with the exceptions of GWCS with RoBERTa-L and WiC with ALBERT-XXL (the former is also an exception observed in the sense profile ablation on §8.1).

As future work, we will also explore WordNet-independent procedures to discover monosemous words, such as the method introduced by Soler & Apidianaki [108], which may lead to further improvements.

---

[11] Due to leaderboard submission limits, ablations for WiC use the validation set.

Table 19

Impact of sense annotations on task performance. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

Task (Metric)	Annotations	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
WSD	SemCor	75.0	<u>74.1</u>	<u>75.2</u>	75.4
(F1 on ALL)	SemCor+UWA ⋆	<u>75.2</u>	<u>74.1</u>	<u>75.2</u>	**<u>75.5</u>**
USM	SemCor	76.3	76.5	76.1	77.4
(P@5 on ALL)	SemCor+UWA ⋆	<u>86.7</u>	<u>87.3</u>	<u>86.9</u>	**<u>87.6</u>**
WiC	SemCor	71.2	67.1	68.5	<u>69.1</u>
(ACC on Val.)	SemCor+UWA ⋆	**<u>71.8</u>**	<u>67.9</u>	<u>68.8</u>	68.7
GWCS	SemCor	76.1	**<u>78.7</u>**	<u>76.3</u>	72.7
(COR on ST2)	SemCor+UWA ⋆	<u>76.3</u>	**<u>78.7</u>**	75.7	<u>75.2</u>
SID	SemCor	72.1	75.2	65.3	73.5
(COR on ALL)	SemCor+UWA ⋆	<u>77.8</u>	**<u>79.5</u>**	<u>74.1</u>	<u>77.4</u>

Table 20

Impact of merging gloss representations on task performance. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

Task (Metric)	Glosses	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
	Without	74.6	<u>74.3</u>	<u>75.3</u>	**<u>75.5</u>**
WSD	Exclusively	57.1	55.2	55.1	54.3
(F1 on ALL)	Averaged ⋆	75.2	74.1	75.2	**<u>75.5</u>**
	Concatenated	**<u>75.5</u>**	<u>74.3</u>	<u>75.3</u>	74.8
	Without	83.5	83.9	83.7	84.2
USM	Exclusively	46.4	44.6	40.5	43.4
(P@5 on ALL)	Averaged ⋆	<u>86.7</u>	<u>87.3</u>	<u>86.9</u>	**<u>87.6</u>**
	Concatenated	85.0	85.7	85.8	86.1
	Without	66.8	64.6	68.7	67.1
WiC	Exclusively	66.3	62.2	66.8	64.1
(ACC on Val.)	Averaged ⋆	**<u>71.8</u>**	<u>67.9</u>	<u>68.8</u>	<u>68.7</u>
	Concatenated	69.3	66.5	68.5	67.7
	Without	75.6	77.3	75.0	74.4
GWCS	Exclusively	75.1	72.9	75.0	68.6
(COR on ST2)	Averaged ⋆	<u>76.3</u>	**<u>78.7</u>**	75.7	<u>75.2</u>
	Concatenated	75.7	77.2	<u>75.8</u>	74.8
	Without	69.5	72.5	62.0	68.3
SID	Exclusively	68.3	70.0	65.1	65.9
(COR on ALL)	Averaged ⋆	<u>77.8</u>	**<u>79.5</u>**	<u>74.1</u>	<u>77.4</u>
	Concatenated	76.7	77.9	69.9	73.8

8.3. Merging gloss representations

Another aspect of LMMS-SP that differs from LMMS [61] is merging gloss embeddings by averaging with sense embeddings, instead of through concatenation (described in Section 4.2.2).

Results on Table 20 reveal that concatenation only benefits WSD, with minor improvements over averaging. Alternatively, averaging shows clear improvements for all other tasks (again, the exception is GWCS w/RoBERTa-L).

We also report performance using exclusively gloss representations, and sense embeddings without gloss information. Surprisingly, these two sets of results are very close on WiC, GWCS and SID, showing that unsupervised representations learned from glosses can be competitive on particular tasks.

8.4. Learning synsets directly

The SID task, as well as the synset version of USM, require synset-level embeddings. In Section 4.1, we explain that LMMS-SP synset embeddings are learned directly from sensekey annotations that are converted to synsets. However, in our evaluation we compare LMMS-SP with other works that are only available as sensekey embeddings, so we converted these representations into synset embeddings learned as the average of corresponding sensekey embeddings (i.e., learned indirectly).

Table 21

Impact of learning synset representations directly from annotations, or indirectly as the average of corresponding sensekey embeddings. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

Task (Metric)	Synset Repr.	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
USM	Indirect	74.5	76.1	77.3	76.3
(P@5 on ALL)	Direct ⋆	<u>81.7</u>	**82.7**	<u>80.1</u>	<u>81.9</u>
SID	Indirect	77.3	78.6	73.0	75.5
(COR on ALL)	Direct ⋆	<u>77.8</u>	**79.5**	<u>74.1</u>	<u>77.4</u>

Table 22

Performance on the combined set of Raganato et al. [94], grouped by part-of-speech. Reporting F1 for WSD and P@5 for USM. MFS not applicable for USM.

Model	Nouns		Verbs		Adjectives		Adverbs	
	WSD	USM	WSD	USM	WSD	USM	WSD	USM
MFS	67.6	N/A	49.6	N/A	78.3	N/A	80.5	N/A
LMMS$_{1024}$ (2019)	75.6	48.2	63.6	65.3	79.8	75.6	**85.0**	78.6
LMMS$_{2048}$ (2019)	78.0	54.3	64.0	64.6	**80.7**	74.0	83.5	77.2
LMMS-SP$_{BERT-L}$	78.0	87.2	63.0	84.2	80.3	85.3	83.8	96.5
LMMS-SP$_{XLNet-L}$	76.8	**87.5**	63.3	86.6	76.9	**86.8**	**85.0**	96.5
LMMS-SP$_{RoBERTa-L}$	**78.2**	86.9	63.1	**86.8**	79.2	86.1	84.7	96.2
LMMS-SP$_{ALBERT-XXL}$	77.8	87.3	**65.6**	86.6	79.1	86.7	84.1	**97.1**

On Table 21 we compare LMMS-SP embeddings learned directly and indirectly, showing that learning these representations directly leads to an average improvement across models of 7.3% on USM, and 1.5% on SID. The fact that indirect representation of synsets has a reduced impact on SID performance, in comparison to USM, suggests that indirect representation leads to more intermingled synset embeddings (i.e., harder to rank), but nearly as globally coherent as those learned from direct representation.

8.5. Part-of-speech performance

In Loureiro & Jorge [61] we presented an error analysis targeting part-of-speech mismatch between predicted and ground-truth senses, which showed that verbs were particularly challenging. In this work, we complement those results by reporting performance by part-of-speech, while comparing LMMS [61] with LMMS-SP.

Our results on Table 22 confirm that verbs remain the most challenging part-of-speech to disambiguate correctly, although ALBERT-XXL shows appreciably better verb results than the other NLMs used in this work. Considering ranked USM matches, however, we find a much narrower gap between verbs and other parts-of-speech using LMMS-SP, with verbs performing comparably with adjectives and nouns, and only BERT-L showing differences larger than 1%.

It is also interesting to note that XLNet-L outperforms or equals ALBERT-XXL on USM for all parts-of-speech with the exception of adverbs, providing better insight into the overall performance differences reported in USM evaluation (§7.2), where ALBERT-XXL outperforms XLNet-L.

9. Discussion

In this section we discuss the main findings of our work. More specifically, we discuss sense representation at specific layers of NLMs (§9.1), differences observed across models and variants (§9.2), and finally, how our sense embeddings may benefit downstream tasks (§9.3).

9.1. Layer distribution

Throughout this article we have provided empirical evidence supporting that there is substantial non-monotonic variation in the adeptness of specific layers of Transformer-based NLMs for sense representation. This evidence is available from both our probing analysis and the improvements in several sense-related tasks obtained from using our proposed sense profiles, which are based on non-monotonic pooling from all layers (most clearly shown in Table 3).

The cause for this variation remains elusive, calling for controlled experiments where different NLMs are tested under comparable circumstances, particularly with regards to training data and modeling objectives, although such an experimental setup may be cost-prohibitive for models of this scale. Nevertheless, seeking to better understand this variation, we conducted two qualitative experiments targeting representations of the same sentences at different layers.

In our first qualitative experiment, we compared sense similarity at different layers for the same word in context. We found some evidence potentially in support of the hypothesis advanced by Voita et al. [117], with the distribution of final
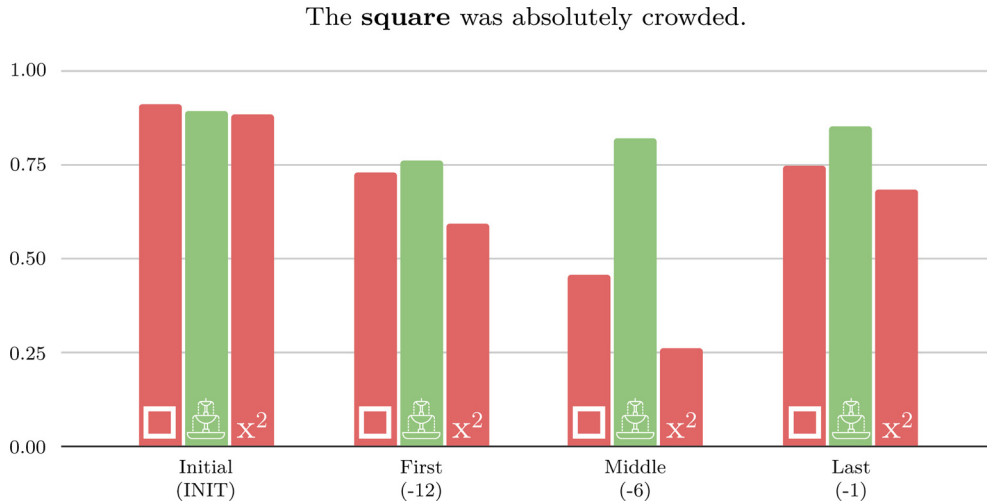
The **square** was absolutely crowded.

Fig. 2. Cosine similarity at specific layers for the word 'square' in context and the 3 senses annotated in SemCor, represented with ALBERT-XXL. The 3 senses of square correspond to the shape, public square (correct, green bar chart) and mathematical operation, in order. Initial layer similarities are influenced by the word forms used with each sense in SemCor.

Table 23
Mean silhouette scores for all 20 words of the 10-shot training instances (balanced) of CoarseWSD-20 [63]. Top rows report scores for specific layers, bottom rows report scores when pooling the sum of last 4 layers and our proposed pooling strategies.

Pooling	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
First Layer	0.156	0.064	0.010	0.137
Middle Layer	0.384	0.167	0.180	0.328
Final Layer	0.369	0.049	0.210	0.273
Sum Last 4	0.376	0.088	**0.218**	0.347
SP-WSD	0.377	0.250	0.203	0.387
SP-USM	**0.388**	**0.255**	0.196	**0.390**

layers resembling the distribution of the first layers moreso than the distribution of middlemost layers, where the difference between correct and incorrect senses is more marked (see example in Fig. 2).

We further extended the previous experiment to a cluster-level comparison of the embedding space. For this experiment, we focus on the words present in the CoarseWSD-20 dataset [63], both in aggregate for measuring correct sense clustering, as well as targeting "spring" and its three distinct senses for visualization. Considering silhouette scores[12] [100] and PCA visualizations of the embedding space (Table 23 and Fig. 3), we arrived at similar conclusions, namely that final layers tend to produce less accurate representations than layers closer to the middle, while the first layer show lowest scores. Our proposed layer pooling methods also show generally improved clustering in comparison to the sum of the last four layers. In addition, this experiment further confirms the unexpected finding regarding a different pattern of semantic representation across layers for XLNet-L, with representations from its final layer showing atypical dispersion.

We leave a more thorough large-scale analysis of this phenomenon for future work, alongside how to appropriately account for measuring the granularity of the different senses of a word, among other confounding factors.

9.2. NLM idiosyncrasies

Besides unexpected results regarding the performance of particular layers of NLMs, we also find intriguing differences in the patterns of layer performance observed across models, and even variants of the same model. Looking at our results on Table 3, we find many intriguing examples of this variation.

For the WSD task, the most striking examples are the differences between BERT-L-UNC-WHL and any other BERT model, and the bi-modal distribution for XLNet. For example, XLNet-B exhibited its best-performing layer near the top of the model, while the best-performing layer for XLNet-L is in the bottom-half of the model. While results for USM are more consistent,

---

[12] We use the mean silhouette coefficients of all embeddings for a particular word to measure how well each model and pooling strategy can assign embeddings to the correct sense cluster. Silhouette coefficients are based on intra- and nearest-cluster cosine similarities. Low values represent overlapping clusters.
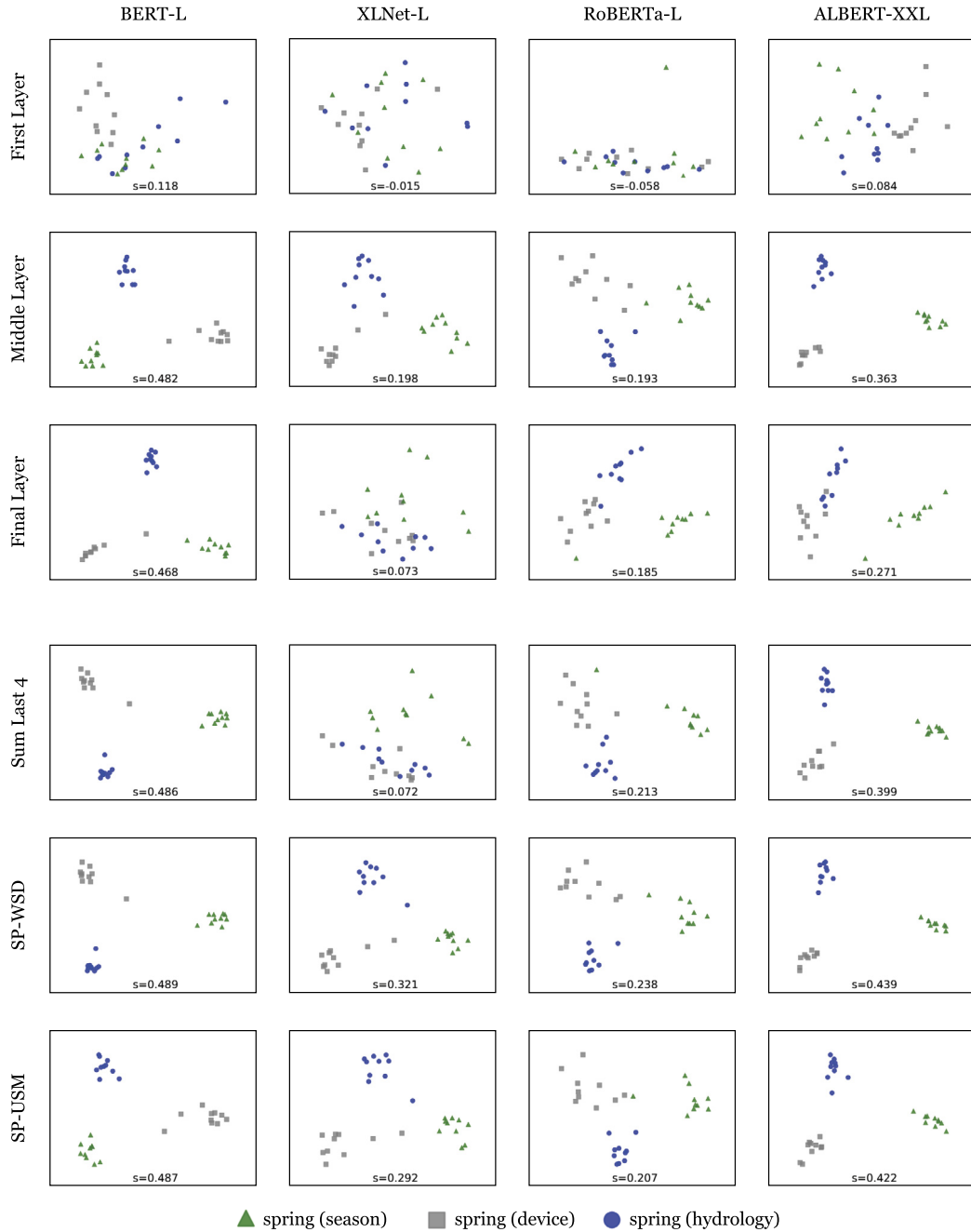
Fig. 3. Visualization of embedding spaces using different pooling strategies. The last two rows correspond to our proposed pooling strategies (see §4.4 and §6). Each point corresponds to an embedding for the word "spring" in context, as provided in the 10-shot set of CoarseWSD-20 [63]. Using PCA for dimensionality reduction. Silhouette scores s are computed before reduction.

we also find some peculiarities there, such as XLNet models showing worst-performing layers at the top, and ALBERT-XL showing a more biased distribution than other ALBERT variants.

The reasons for these differences in patterns across models and variants are not straightforward, specially considering many of these models are trained on similar data and architectures. Still, among several technical differences, we highlight the differences in modeling objectives covered in Section 5.1. Out of the 4 the models we considered in this work (see performance summary on Table 24), XLNet is in fact simultaneously the model that appears most distinctive, with particularly strong performance on graded similarity tasks, and whose objectives are most different (being the only model not using MLM). Another interesting finding is that we obtain best results on WSD, USM and WiC using ALBERT-XXL, which has half the layers of the other models, but much larger embedding dimensionality (model details are available in Table 2). As for

Table 24

Summary comparison between different NLMs using our LMMS-SP approach.

Model	WSD (F1)	USM (P@5)	WiC (ACC)	GWCS (COR)	SID (COR)
LMMS-SP$_{BERT-L}$	75.2	86.7	67.4	76.3	77.8
LMMS-SP$_{XLNet-L}$	74.1	87.3	66.1	**78.7**	**79.5**
LMMS-SP$_{RoBERTa-L}$	75.2	86.9	67.8	75.7	74.1
LMMS-SP$_{ALBERT-XXL}$	**75.5**	**87.6**	**67.9**	75.2	77.4
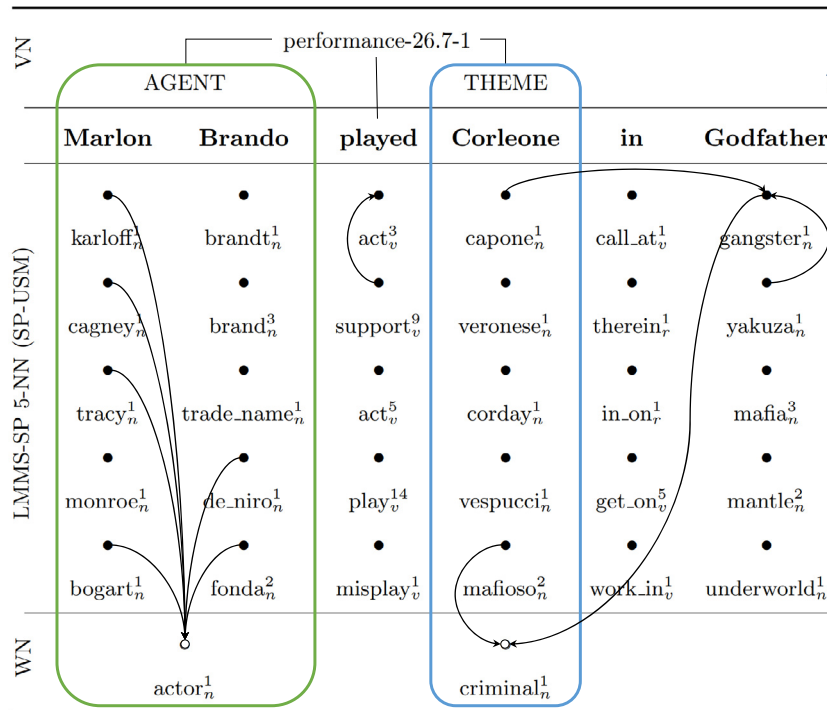
Fig. 4. Example sentence with each token matched to LMMS-SP$_{ALBERT-XXL}$ sense embeddings, presenting synsets for the 5 nearest neighbors, using the SP-USM sense profile. Shows direct hypernymy relations (i.e., Is-A), included in WordNet (WN), between matched synsets, as well as hypernymy relations shared between more than one matched and unmatched synset (i.e., deducible generalizations, not in top 5 matches). Finally, at the top, we show a VerbNet (VN) semantic frame matched to this sentence, highlighting how LMMS-SP enables generalization of argument spans.

differences in variants of the same model (same objectives) we consider the possibility that trivial run-time parameters may have an impact on this variation, akin to the unexpected influence of random seeds on fine-tuning BERT models [30].

9.3. Knowledge integration

The ability of matching WordNet synsets to any fragment of text allows downstream applications to easily leverage the manually curated relations available on WordNet. At the same time, these sense embeddings can also serve as an entry point to many other knowledge bases linked to WordNet, such as the multilingual knowledge graph of BabelNet [76], the common-sense triples of ConceptNet [109] or WebChild [110], the semantic frames of VerbNet [106], and even the images of ImageNet [101] or Visual Genome [49]. Several recent works have used the symbolic relations expressed in these knowledge bases to improve neural solutions to Natural Language Inference [47], Commonsense Reasoning [57], Story Generation [1], among others.

As an example of how using LMMS-SP to bridge natural language and symbolic knowledge can be beneficial, in Fig. 4 we demonstrate how these sense embeddings allow for generalization of argument spans, predicted by a semantic parser, exploiting WordNet relations between matched synsets. The matches shown in Fig. 4 also illustrate how sense embeddings may be used for probing world knowledge encoded in pre-trained NLMs, as already suggested in Loureiro & Jorge [61].

10. Conclusion

Leveraging neural language models in combination with sense-annotated corpora (and complementary resources such as glosses or relations), this work has shown that it is possible to produce sense embeddings applicable beyond mere disambiguation, with relevant implications for long-standing challenges in Artificial Intelligence such as symbol grounding.

This extension of Loureiro & Jorge [61] proposes a more principled approach for learning distributional representations of word senses using pre-trained NLMs, focusing on state-of-the-art Transformer-based models. From extensive evaluation on several sense-related tasks, we demonstrated that the LMMS-SP approach is more effective than prior work at approximating precise word sense representations in the same vector space of NLMs.

The broad probing analysis of the many variants of popular NLMs endeavored in this work provides new evidence supporting further research on the interplay between pre-training objectives, layer specialization, and model size. The conclusions of this probing analysis are indeed expected to be applicable in tasks outside WSD, and for learning representations other than sense embeddings, which we leave for future work.

Effectively, there are known limitations to meaning representation based on language modeling objectives alone [6,69]. Nonetheless, we believe our work shows there is still much to understand about how to best leverage NLMs for meaning representation, in addition to more thoroughly testing the effectiveness of current approaches centered on self-supervision.

Release. This work is accompanied by the release of the following resources: sensekey and synset embeddings with full-coverage of WordNet based on BERT-L, XLNet-L, RoBERTa-L and ALBERT-XXL; scripts to generate embeddings following our method, using the same NLMs or others supported by the Transformers package; and scripts to run task evaluations. These resources are released under a GNU General Public License (v3) and available from this public repository: https://github.com/danlou/lmms

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Appendix A. Full results for SemEval 2020 - Task 3

On Table 25 we report complete leaderboard results for subtasks 1 and 2 of SemEval 2020 Task 3 (including other languages besides English), during the evaluation period.

Table 25
Results from the leaderboard of subtasks 1 and 2 of SemEval 2020 Task 3 - Predicting the (Graded) Effect of Context in Word Similarity. Rank reported in team names. At the time of this evaluation, we did not use the sense profiles proposed in this paper, so our reported results on this table are based on senses embeddings pooled from the last 4 layers of the specified models, following Loureiro & Jorge [61].

English				Hungarian			
Team	Sub1	Team	Sub2	Team	Sub1	Team	Sub2
1. Ferryman	0.774	1. MineriaUNAM	0.723	1. N+S	0.740	1. N+S	0.658
2. will_go	0.768	**2.** LMMS$_{RoBERTa-L}$	0.720	2. Hitachi	0.681	2. Hitachi	0.616
3. MULTISEM	0.760	3. somaia	0.719	3. InfoMiner	0.754	3. MineriaUNAM	0.613
4. LMMS$_{RoBERTa-L}$	0.754	4. MULTISEM	0.718	4. Ferryman	0.774	**4.** LMMS$_{XLMR-L}$	0.565
5. InfoMiner	0.754	5. InfoMiner	0.715	**5.** LMMS$_{XLMR-L}$	0.754	5. InfoMiner	0.545
Finnish				Slovenian			
Team	Sub1	Team	Sub2	Team	Sub1	Team	Sub2
1. will_go	0.772	1. InfoMiner	0.645	1. Hitachi	0.654	1. N+S	0.579
2. Ferryman	0.745	**2.** N+S	0.611	2. InfoMiner	0.648	2. InfoMiner	0.573
3. N+S	0.726	3. MineriaUNAM	0.597	3. N+S	0.646	3. CitiusNLP	0.538
4. RTM	0.671	4. MULTISEM	0.492	4. CitiusNLP	0.624	4. tthhanh	0.516
11. LMMS$_{XLMR-L}$	0.360	**7.** LMMS$_{XLMR-L}$	0.354	**8.** LMMS$_{XLMR-L}$	0.560	**9.** LMMS$_{XLMR-L}$	0.483

Appendix B. Stanford contextual word similarities (SCWS)

On Table 26 we report our results on the Stanford Contextual Word Similarities (SCWS) [43] task. We address this task similarly to GWCS (see Section 7.4). Given two words in context, each within an independent sentence, we disambiguate both occurrences and score each pair as the average of similarities between corresponding sense and contextual embeddings.

Results on SCWS follow performance on GWCS, with XLNet-L outperforming other NLMs as well as results from related works. Analyzing performance by Part-of-Speech (POS), we find that nouns appear most challenging for this task, particularly when being compared against other nouns.

Table 26
Results on SCWS (Spearman correlation scores, $\rho \times 100$), considering the entire set of pairs (ALL) as well as results for subsets pairing particular Parts-of-Speech (with n denoting the number of instances for each subset), similarly to Colla et al. [20].

System	ALL (n=2003)	N-N (n=1328)	N-V (n=140)	N-A (n=30)	V-V (n=399)	V-A (n=9)	A-A (n=97)
Huang et al. [43]	65.7	–	–	–	–	–	–
SensEmbed [45]	62.4	–	–	–	–	–	–
NASARI [18]	–	47.1	–	–	–	–	–
DeConf [90]	71.5	–	–	–	–	–	–
LessLex [20]	69.5	69.2	69.6	82.0	64.1	73.6	63.8
ARES [105]	67.9	66.6	68.6	**87.9**	67.2	66.7	69.4
BERT-L (SP-WSD)	59.3	56.8	67.4	78.4	59.4	60.0	61.1
XLNet-L (SP-WSD)	73.9	71.6	75.6	81.3	75.8	**78.3**	76.0
RoBERTa-L (SP-WSD)	63.8	59.1	71.3	66.6	68.7	73.3	66.7
ALBERT-XXL (SP-WSD)	65.9	63.7	69.4	74.9	66.3	75.0	69.5
LMMS-SP$_{\text{BERT-L}}$	64.1	62.3	67.1	82.6	63.5	51.7	68.3
LMMS-SP$_{\text{XLNet-L}}$	**75.9**	**73.7**	**75.8**	81.5	**78.0**	75.0	**79.4**
LMMS-SP$_{\text{RoBERTa-L}}$	67.4	63.4	73.9	70.8	71.1	75.0	68.1
LMMS-SP$_{\text{ALBERT-XXL}}$	69.9	68.8	72.4	76.4	69.9	66.6	70.9

References

[1] P. Ammanabrolu, E. Tien, W. Cheung, Z. Luo, W. Ma, L.J. Martin, M.O. Riedl, Story realization: expanding plot events into sentences, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 7375–7382, https://ojs.aaai.org/index.php/AAAI/article/view/6232.

[2] C.S. Armendariz, M. Purver, S. Pollak, N. Ljubešić, M. Ulčar, I. Vulić, M.T. Pilehvar, SemEval-2020 task 3: graded word similarity in context, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona, 2020, pp. 36–49, https://www.aclweb.org/anthology/2020.semeval-1.3.

[3] C.S. Armendariz, M. Purver, M. Ulčar, S. Pollak, N. Ljubešić, M. Granroth-Wilding, CoSimLex: a resource for evaluating graded word similarity in context, in: Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, European Language Resources Association, 2020, pp. 5878–5886, https://www.aclweb.org/anthology/2020.lrec-1.720.

[4] B. Athiwaratkun, A. Wilson, A. Anandkumar, Probabilistic FastText for multi-sense word embeddings, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1–11, https://www.aclweb.org/anthology/P18-1001.

[5] E. Barba, L. Procopio, R. Navigli, ConSeC: word sense disambiguation as continuous sense comprehension, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 1492–1503, https://aclanthology.org/2021.emnlp-main.112.

[6] E.M. Bender, A. Koller, Climbing towards NLU: on meaning, form, and understanding in the age of data, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5185–5198, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.acl-main.463.

[7] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[8] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[9] M. Bevilacqua, R. Navigli, Breaking through the 80% glass ceiling: raising the state of the art in word sense disambiguation by incorporating knowledge graph information, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2854–2864, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.acl-main.255.

[10] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[11] T. Blevins, L. Zettlemoyer, Moving down the long tail of word sense disambiguation with gloss informed bi-encoders, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1006–1017, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.acl-main.95.

[12] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146, https://doi.org/10.1162/tacl_a_00051, https://www.aclweb.org/anthology/Q17-1010.

[13] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR, arXiv:2005.14165 [abs], arXiv:2005.14165, 2020.

[14] Z.G. Cai, R.A. Gilbert, M.H. Davis, M.G. Gaskell, L. Farrar, S. Adler, J.M. Rodd, Accent modulates access to word meaning: evidence for a speaker-model account of spoken word recognition, Cogn. Psychol. 98 (2017) 73–101, https://doi.org/10.1016/j.cogpsych.2017.08.003, http://www.sciencedirect.com/science/article/pii/S0010028517300762.

[15] J. Camacho-Collados, M.T. Pilehvar, From word to sense embeddings: a survey on vector representations of meaning, J. Artif. Intell. Res. 63 (2018) 743–788, https://doi.org/10.1613/jair.1.11259.

[16] J. Camacho-Collados, M.T. Pilehvar, N. Collier, R. Navigli, SemEval-2017 task 2: multilingual and cross-lingual semantic word similarity, in: Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval-2017, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 15–26, https://www.aclweb.org/anthology/S17-2002.

[17] J. Camacho-Collados, M.T. Pilehvar, R. Navigli, NASARI: a novel approach to a semantically-aware representation of items, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 567–577, https://www.aclweb.org/anthology/N15-1059.

[18] J. Camacho-Collados, M.T. Pilehvar, R. Navigli, Nasari: integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities, Artif. Intell. 240 (2016) 36–64, https://doi.org/10.1016/j.artint.2016.07.005, http://www.sciencedirect.com/science/article/pii/S0004370216300820.

[19] G. Chronis, K. Erk, When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships, in: Proceedings of the 24th Conference on Computational Natural Language Learning, 2020, pp. 227–244, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.conll-1.17.

[20] D. Colla, E. Mensa, D.P. Radicioni, LessLex: linking multilingual embeddings to SenSe representations of LEXical items, Comput. Linguist. 46 (2020) 289–333, https://doi.org/10.1162/coli_a_00375, https://www.aclweb.org/anthology/2020.cl-2.3.

[21] D. Colla, E. Mensa, D.P. Radicioni, Novel metrics for computing semantic similarity with sense embeddings, Knowl.-Based Syst. 206 (2020) 106346, https://doi.org/10.1016/j.knosys.2020.106346, https://www.sciencedirect.com/science/article/pii/S0950705120305025.

[22] D. Colla, E. Mensa, D.P. Radicioni, Sense identification data: a dataset for lexical semantics, Data Brief 32 (2020) 106267, https://doi.org/10.1016/j.dib.2020.106267, https://www.sciencedirect.com/science/article/pii/S2352340920311616.

[23] R. Collobert, J. Weston, Fast semantic extraction using a novel neural network architecture, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 560–567, https://www.aclweb.org/anthology/P07-1071.

[24] R. Collobert, J. Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 160–167.

[25] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.

[26] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, R. Salakhutdinov, Transformer-XL: attentive language models beyond a fixed-length context, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2978–2988, https://www.aclweb.org/anthology/P19-1285.

[27] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (1990) 391–407.

[28] S.C. Deerwester, S.T. Dumais, G.W. Furnas, R.A. Harshman, T.K. Landauer, K.E. Lochbaum, L.A. Streeter, Computer information retrieval using latent semantic structure, 1989, US Patent 4,839,853.

[29] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, in: Long and Short Papers, vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, https://www.aclweb.org/anthology/N19-1423.

[30] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, N. Smith, Fine-tuning pretrained language models: weight initializations, data orders, and early stopping, arXiv:2002.06305, 2020.

[31] Z. Dong, Q. Dong, C. Hao, Hownet and the computation of meaning, 2006.

[32] K. Erk, What do you know about an alligator when you know the company it keeps?, Semant. Pragmat. 9 (2016) 1–63, https://doi.org/10.3765/sp.9.17.

[33] K. Ethayarajh, How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 55–65, https://www.aclweb.org/anthology/D19-1006.

[34] C. Fellbaum, Wordnet: An Electronic Lexical Database, MIT Press, 1998.

[35] J. Firth, A synopsis of linguistic theory 1930-1955, in: Studies in Linguistic Analysis, Philological Society, Oxford, 1957, Reprinted in Palmer, F. (ed. 1968) Selected Papers of J.R. Firth, Longman, Harlow.

[36] J.R. Firth, The technique of semantics, Trans. Philol. Soc. 34 (1935) 36–73.

[37] L. Flekova, I. Gurevych, Supersense embeddings: a unified model for supersense interpretation, prediction, and utilization, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2029–2041, https://www.aclweb.org/anthology/P16-1191.

[38] Y. Goldberg, Neural network methods for natural language processing, Synth. Lect. Hum. Lang. Technol. 10 (2017) 1–309.

[39] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: PMLR Proceedings of Machine Learning Research, vol. 70, 2017, pp. 1321–1330, http://proceedings.mlr.press/v70/guo17a.html.

[40] W.L. Hamilton, J. Leskovec, D. Jurafsky, Diachronic word embeddings reveal statistical laws of semantic change, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1489–1501, https://www.aclweb.org/anthology/P16-1141.

[41] Z.S. Harris, Distributional structure, Word 10 (1954) 146–162.

[42] J. Hewitt, C.D. Manning, A structural probe for finding syntax in word representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, in: Long and Short Papers, vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4129–4138, https://www.aclweb.org/anthology/N19-1419.

[43] E. Huang, R. Socher, C. Manning, A. Ng, Improving word representations via global context and multiple word prototypes, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 873–882, https://www.aclweb.org/anthology/P12-1092.

[44] L. Huang, C. Sun, X. Qiu, X. Huang, GlossBERT: BERT for word sense disambiguation with gloss knowledge, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3509–3514, https://www.aclweb.org/anthology/D19-1355.

[45] I. Iacobacci, M.T. Pilehvar, R. Navigli, SensEmbed: learning sense embeddings for word and relational similarity, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, in: Long Papers, vol. 1, Association for Computational Linguistics, Beijing, China, 2015, pp. 95–105, https://www.aclweb.org/anthology/P15-1010.

[46] N. Ide, C.F. Baker, C. Fellbaum, R.J. Passonneau, The manually annotated sub-corpus: a community resource for and by the people, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Short Papers, Uppsala, Sweden, 2010, pp. 68–73.

[47] P. Kapanipathi, V. Thost, S. Sankalp Patel, S. Whitehead, I. Abdelaziz, A. Balakrishnan, M. Chang, K. Fadnis, C. Gunasekara, B. Makni, N. Mattei, K. Talamadupula, A. Fokoue, Infusing knowledge into the textual entailment task using graph convolutional networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 8074–8081, https://ojs.aaai.org/index.php/AAAI/article/view/6318.

[48] D.E. Klein, G.L. Murphy, The representation of polysemous words, J. Mem. Lang. 45 (2001) 259–282, https://doi.org/10.1006/jmla.2001.2779, http://www.sciencedirect.com/science/article/pii/S0749596X01927792.

[49] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, M. Bernstein, L. Fei-Fei, in: Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations, 2016, https://arxiv.org/abs/1602.07332.

[50] T. Kudo, J. Richardson, SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 66–71, https://www.aclweb.org/anthology/D18-2012.

[51] I. Kuznetsov, I. Gurevych, A matter of framing: the impact of linguistic formalism on probing results, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 171–182, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.emnlp-main.13.

[52] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: a lite bert for self-supervised learning of language representations, in: International Conference on Learning Representations, 2020, https://openreview.net/forum?id=H1eA7AEtvS.

[53] T.K. Landauer, S.T. Dumais, A solution to plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge, Psychol. Rev. 104 (1997) 211.

[54] Y. Levine, B. Lenz, O. Dagan, O. Ram, D. Padnos, O. Sharir, S. Shalev-Shwartz, A. Shashua, Y. Shoham, SenseBERT: driving some sense into BERT, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4656–4667, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.acl-main.423.

[55] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.acl-main.703.

[56] J. Li, D. Jurafsky, Do multi-sense embeddings improve natural language understanding?, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1722–1732, https://www.aclweb.org/anthology/D15-1200.

[57] B.Y. Lin, X. Chen, J. Chen, X. Ren, KagNet: knowledge-aware graph networks for commonsense reasoning, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2829–2839, https://www.aclweb.org/anthology/D19-1282.

[58] N.F. Liu, M. Gardner, Y. Belinkov, M.E. Peters, N.A. Smith, Linguistic knowledge and transferability of contextual representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, in: Long and Short Papers, vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1073–1094, https://www.aclweb.org/anthology/N19-1112.

[59] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: a robustly optimized BERT pretraining approach, CoRR, arXiv:1907.11692 [abs], 2019.

[60] D. Loureiro, J. Camacho-Collados, Don't neglect the obvious: on the role of unambiguous words in word sense disambiguation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 3514–3520, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.emnlp-main.283.

[61] D. Loureiro, A. Jorge, 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, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5682–5691, https://www.aclweb.org/anthology/P19-1569.

[62] D. Loureiro, A. Jorge, LIAAD at SemDeep-5 challenge: Word-in-Context (WiC), in: Proceedings of the 5th Workshop on Semantic Deep Learning, SemDeep-5, Association for Computational Linguistics, Macau, China, 2019, pp. 1–5, https://www.aclweb.org/anthology/W19-5801.

[63] D. Loureiro, K. Rezaee, M.T. Pilehvar, J. Camacho-Collados, Analysis and evaluation of language models for word sense disambiguation, Comput. Linguist. (2021) 1–55, https://doi.org/10.1162/coli_a_00405.

[64] K. Lund, C. Burgess, Producing high-dimensional semantic spaces from lexical co-occurrence, Behav. Res. Methods Instrum. Comput. 28 (1996) 203–208.

[65] M. Mancini, J. Camacho-Collados, I. Iacobacci, R. Navigli, Embedding words and senses together via joint knowledge-enhanced training, in: Proceedings of the 21st Conference on Computational Natural Language Learning, CoNLL 2017, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 100–111, https://www.aclweb.org/anthology/K17-1012.

[66] T. McCoy, E. Pavlick, T. Linzen, Right for the wrong reasons: diagnosing syntactic heuristics in natural language inference, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3428–3448, https://www.aclweb.org/anthology/P19-1334.

[67] S. Mcdonald, M. Ramscar, Testing the distributional hypothesis: the influence of context on judgements of semantic similarity, in: Proceedings of the 23rd Annual Conference of the Cognitive Science Society, 2001, pp. 611–616.

[68] O. Melamud, J. Goldberger, I. Dagan, context2vec: learning generic context embedding with bidirectional LSTM, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 51–61, https://www.aclweb.org/anthology/K16-1006.

[69] W. Merrill, Y. Goldberg, R. Schwartz, N.A. Smith, Provable limitations of acquiring meaning from ungrounded form: what will future language models understand?, arXiv:2104.10809, 2021.

[70] C.M. Meyer, I. Gurevych, Wiktionary: a new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography, 2012.

[71] T. Mickus, D. Paperno, M. Constant, K. van Deemter, What do you mean, bert? Assessing bert as a distributional semantics model, in: Proceedings of the Society for Computation in Linguistics, vol. 3, 2020.

[72] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, pp. 3111–3119.

[73] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[74] G.A. Miller, M. Chodorow, S. Landes, C. Leacock, R.G. Thomas, Using a semantic concordance for sense identification, in: Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 8-11, 1994, 1994, https://www.aclweb.org/anthology/H94-1046.

[75] R. Navigli, Word sense disambiguation: a survey, ACM Comput. Surv. 41 (2009) 10:1–10:69, https://doi.org/10.1145/1459352.1459355, http://doi.acm.org/10.1145/1459352.1459355.

[76] R. Navigli, S.P. Ponzetto, in: BabelNet: Building a Very Large Multilingual Semantic Network, 2010, pp. 216–225.

[77] A. Neelakantan, J. Shankar, A. Passos, A. McCallum, Efficient non-parametric estimation of multiple embeddings per word in vector space, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1059–1069, https://www.aclweb.org/anthology/D14-1113.

[78] C.E. Osgood, G.J. Suci, P.H. Tannenbaum, The Measurement of Meaning, vol. 47, University of Illinois Press, 1957.

[79] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, M. Auli, fairseq: a fast, extensible toolkit for sequence modeling, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Demonstrations, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 48–53, https://www.aclweb.org/anthology/N19-4009.

[80] T. Pasini, The knowledge acquisition bottleneck problem in multilingual word sense disambiguation, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-20, Yokohama, Japan, 2020.

[81] M. Pelevina, N. Arefiev, C. Biemann, A. Panchenko, Making sense of word embeddings, in: Proceedings of the 1st Workshop on Representation Learning for NLP, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 174–183, https://www.aclweb.org/anthology/W16-1620.

[82] J. Pennington, R. Socher, C. Manning, GloVe: global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543, https://www.aclweb.org/anthology/D14-1162.

[83] F. Pereira, B. Lou, B. Pritchett, S. Ritter, S.J. Gershman, N. Kanwisher, M. Botvinick, E. Fedorenko, Toward a universal decoder of linguistic meaning from brain activation, Nat. Commun. 9 (2018) 1–13.

[84] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, in: Long Papers, vol. 1, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237, https://www.aclweb.org/anthology/N18-1202.

[85] M. Peters, M. Neumann, L. Zettlemoyer, W.-t. Yih, Dissecting contextual word embeddings: architecture and representation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1499–1509, https://www.aclweb.org/anthology/D18-1179.

[86] M.E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, N.A. Smith, Knowledge enhanced contextual word representations, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 43–54, https://www.aclweb.org/anthology/D19-1005.

[87] S.T. Piantadosi, H. Tily, E. Gibson, The communicative function of ambiguity in language, Cognition 122 (2012) 280–291, https://doi.org/10.1016/j.cognition.2011.10.004, http://www.sciencedirect.com/science/article/pii/S0010027711002496.

[88] M.T. Pilehvar, J. Camacho-Collados, WiC: the word-in-context dataset for evaluating context-sensitive meaning representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, in: Long and Short Papers, vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1267–1273, https://www.aclweb.org/anthology/N19-1128.

[89] M.T. Pilehvar, J. Camacho-Collados, R. Navigli, N. Collier, Towards a seamless integration of word senses into downstream NLP applications, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1857–1869, https://www.aclweb.org/anthology/P17-1170.

[90] M.T. Pilehvar, N. Collier, De-conflated semantic representations, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 1680–1690, https://www.aclweb.org/anthology/D16-1174.

[91] R. Radach, H. Deubel, C. Vorstius, M. Hofmann (Eds.), Abstracts of the 19th European Conference on Eye Movements, 2017, J. Eye Mov. Res. 10 (2017), https://doi.org/10.16910/jemr.10.6.1, https://bop.unibe.ch/JEMR/article/view/JEMR.10.6.1.

[92] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019.

[93] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 1–67, http://jmlr.org/papers/v21/20-074.html.

[94] A. Raganato, J. Camacho-Collados, R. Navigli, Word sense disambiguation: a unified evaluation framework and empirical comparison, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 99–110, https://www.aclweb.org/anthology/E17-1010.

[95] E. Reif, A. Yuan, M. Wattenberg, F.B. Viegas, A. Coenen, A. Pearce, B. Kim, Visualizing and measuring the geometry of bert, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlche Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, pp. 8594–8603, https://proceedings.neurips.cc/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf.

[96] J. Reisinger, R.J. Mooney, Multi-prototype vector-space models of word meaning, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 109–117, https://www.aclweb.org/anthology/N10-1013.

[97] J.M. Rodd, Settling into semantic space: an ambiguity-focused account of word-meaning access, Perspectives Psychol. Sci. 15 (2020) 411–427, https://doi.org/10.1177/1745691619885860, PMID: 31961780.

[98] A. Rogers, O. Kovaleva, A. Rumshisky, A primer in BERTology: what we know about how BERT works, Trans. Assoc. Comput. Linguist. 8 (2020) 842–866, https://doi.org/10.1162/tacl_a_00349, https://www.aclweb.org/anthology/2020.tacl-1.54.

[99] S. Rothe, H. Schütze, AutoExtend: extending word embeddings to embeddings for synsets and lexemes, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, in: Long Papers, vol. 1, Association for Computational Linguistics, Beijing, China, 2015, pp. 1793–1803, https://www.aclweb.org/anthology/P15-1173.

[100] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65, https://doi.org/10.1016/0377-0427(87)90125-7.

[101] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[102] G. Salton, The smart system, in: Retrieval Results and Future Plans, 1971.

[103] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (1975) 613–620.

[104] B. Scarlini, T. Pasini, R. Navigli, SensEmBERT: context-enhanced sense embeddings for multilingual word sense disambiguation, in: Proceedings of the Thirty-Fourth Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence, 2020, pp. 8758–8765.

[105] B. Scarlini, T. Pasini, R. Navigli, With more contexts comes better performance: contextualized sense embeddings for all-round word sense disambiguation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 3528–3539, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.emnlp-main.285.

[106] K.K. Schuler, VerbNet: a broad-coverage, comprehensive verb lexicon, Ph.D. thesis, University of Pennsylvania, 2006, http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf.

[107] H. Schutze, Dimensions of meaning, in: Supercomputing'92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing, IEEE, 1992, pp. 787–796.

[108] A.G. Soler, M. Apidianaki, Let's play mono-poly: bert can reveal words' polysemy level and partitionability into senses, in: Transactions of the Association for Computational Linguistics, TACL, 2021.

[109] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: an open multilingual graph of general knowledge, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, AAAI Press, 2017, pp. 4444–4451.

[110] N. Tandon, G. de Melo, G. Weikum, WebChild 2.0: fine-grained commonsense knowledge distillation, in: Proceedings of ACL 2017, System Demonstrations, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 115–120, https://www.aclweb.org/anthology/P17-4020.

[111] I. Tenney, D. Das, E. Pavlick, BERT rediscovers the classical NLP pipeline, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4593–4601, https://www.aclweb.org/anthology/P19-1452.

[112] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R.T. McCoy, N. Kim, B.V. Durme, S.R. Bowman, D. Das, E. Pavlick, What do you learn from context? Probing for sentence structure in contextualized word representations, in: International Conference on Learning Representations, 2019, https://openreview.net/forum?id=SJzSgnRcKX.

[113] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[114] L. Vial, B. Lecouteux, D. Schwab, UFSAC: unification of sense annotated corpora and tools, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, European Language Resources Association (ELRA), Miyazaki, Japan, 2018, https://www.aclweb.org/anthology/L18-1166.

[115] L. Vial, B. Lecouteux, D. Schwab, Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation, in: Proceedings of the 10th Global Wordnet Conference, Global Wordnet Association, Wroclaw, Poland, 2019, pp. 108–117, https://www.aclweb.org/anthology/2019.gwc-1.14.

[116] E. Voita, R. Sennrich, I. Titov, The bottom-up evolution of representations in the transformer: a study with machine translation and language modeling objectives, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 4396–4406, https://www.aclweb.org/anthology/D19-1448.

[117] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5797–5808, https://www.aclweb.org/anthology/P19-1580.

[118] T. Vu, D.S. Parker, *k*-Embeddings: learning conceptual embeddings for words using context, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1262–1267, https://www.aclweb.org/anthology/N16-1151.

[119] I. Vulić, E.M. Ponti, R. Litschko, G. Glavaš, A. Korhonen, Probing pretrained language models for lexical semantics, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 7222–7240, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.emnlp-main.586.

[120] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Superglue: a stickier benchmark for general-purpose language understanding systems, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlche Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

[121] L. Wittgenstein, Philosophical investigations, trans, GEM Anscombe 261 (1953) 49.

[122] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: state-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[123] Y. Yaghoobzadeh, H. Schütze, Intrinsic subspace evaluation of word embedding representations, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, in: Long Papers, vol. 1, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 236–246, https://www.aclweb.org/anthology/P16-1023.

[124] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, 2019, pp. 5753–5763.

[125] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: 33rd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Cambridge, Massachusetts, USA, 1995, pp. 189–196, https://www.aclweb.org/anthology/P95-1026.

[126] D. Yuan, J. Richardson, R. Doherty, C. Evans, E. Altendorf, Semi-supervised word sense disambiguation with neural models, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 1374–1385, https://www.aclweb.org/anthology/C16-1130.

[127] X. Zhou, M. Sap, S. Swayamdipta, Y. Choi, N. Smith, Challenges in automated debiasing for toxic language detection, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 3143–3155, Online: Association for Computational Linguistics, https://www.aclweb.org/anthology/2021.eacl-main.274.

[128] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: towards story-like visual explanations by watching movies and reading books, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 19–27.