Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation

Daniel Loureiro, Alípio Mário Jorge
LIAAD - INESC TEC
Faculty of Sciences - University of Porto, Portugal
dloureiro@fc.up.pt, amjorge@fc.up.pt

Abstract

Contextual embeddings represent a new generation of semantic representations learned from Neural Language Modelling (NLM) that addresses the issue of meaning conflation hampering traditional word embeddings. In this work, we show that contextual embeddings can be used to achieve unprecedented gains in Word Sense Disambiguation (WSD) tasks. Our approach focuses on creating sense-level embeddings with full-coverage of WordNet, and without recourse to explicit knowledge of sense distributions or task-specific modelling. As a result, a simple Nearest Neighbors (k-NN) method using our representations is able to consistently surpass the performance of previous systems using powerful neural sequencing models. We also analyse the robustness of our approach when ignoring part-of-speech and lemma features, requiring disambiguation against the full sense inventory, and revealing shortcomings to be improved. Finally, we explore applications of our sense embeddings for concept-level analyses of contextual embeddings and their respective NLMs.

1 Introduction

Word Sense Disambiguation (WSD) is a core task of Natural Language Processing (NLP) which consists in assigning the correct sense to a word in a given context, and has many potential applications (Navigli, 2009). Despite breakthroughs in distributed semantic representations (i.e. word embeddings), resolving lexical ambiguity has remained a long-standing challenge in the field. Systems using non-distributional features, such as It Makes Sense (IMS, Zhong and Ng, 2010), remain surprisingly competitive against neural sequence models trained end-to-end. A baseline that simply chooses the most frequent sense (MFS) has also proven to be notoriously difficult to surpass.

Several factors have contributed to this limited progress over the last decade, including lack of standardized evaluation, and restricted amounts of sense annotated corpora. Addressing the evaluation issue, Raganato et al. (2017a) has introduced a unified evaluation framework that has already been adopted by the latest works in WSD. Also, even though SemCor (Miller et al., 1994) still remains the largest manually annotated corpus, supervised methods have successfully used label propagation (Yuan et al., 2016), semantic networks (Vial et al., 2018) and glosses (Luo et al., 2018b) in combination with annotations to advance the state-of-the-art. Meanwhile, task-specific sequence modelling architectures based on BiLSTMs or Seq2Seq (Raganato et al., 2017b) haven't yet proven as advantageous for WSD.

Until recently, the best semantic representations at our disposal, such as word2vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017), were bound to word types (i.e. distinct tokens), converging information from different senses into the same representations (e.g. 'play song' and 'play tennis' share the same representation of 'play'). These word embeddings were learned from unsupervised Neural Language Modelling (NLM) trained on fixed-length contexts. However, by recasting the same word types across different sense-inducing contexts, these representations became insensitive to the different senses of polysemous words. Camacho-Collados and Pilehvar (2018) refer to this issue as the meaning conflation deficiency and explore it more thoroughly in their work.

Recent improvements to NLM have allowed for learning representations that are context-specific and detached from word types. While word embedding methods reduced NLMs to fixed representations after pretraining, this new generation of contextual embeddings employs the pretrained

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5682–5691
Florence, Italy, July 28 - August 2, 2019.

NLM to infer different representations induced by arbitrarily long contexts. Contextual embeddings have already had a major impact on the field, driving progress on numerous downstream tasks. This success has also motivated a number of iterations on embedding models in a short timespan, from context2vec (Melamud et al., 2016), to GPT (Radford et al., 2018), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019).

Being context-sensitive by design, contextual embeddings are particularly well-suited for WSD. In fact, Melamud et al. (2016) and Peters et al. (2018) produced contextual embeddings from the SemCor dataset and showed competitive results on Raganato et al. (2017a)'s WSD evaluation framework, with a surprisingly simple approach based on Nearest Neighbors (k-NN). These results were promising, but those works only produced sense embeddings for the small fraction of WordNet (Fellbaum, 1998) senses covered by SemCor, resorting to the MFS approach for a large number of instances. Lack of high coverage annotations is one of the most pressing issues for supervised WSD approaches (Le et al., 2018).

Our experiments show that the simple k-NN w/MFS approach using BERT embeddings suffices to surpass the performance of all previous systems. Most importantly, in this work we introduce a method for generating sense embeddings with full-coverage of WordNet, which further improves results (additional 1.9% F1) while forgoing MFS fallbacks. To better evaluate the fitness of our sense embeddings, we also analyse their performance without access to lemma or part-of-speech features typically used to restrict candidate senses. Representing sense embeddings in the same space as any contextual embeddings generated from the same pretrained NLM eases introspections of those NLMs, and enables token-level intrinsic evaluations based on k-NN WSD performance. We summarize our contributions[1] below:

- A method for creating sense embeddings for all senses in WordNet, allowing for WSD based on k-NN without MFS fallbacks.
- Major improvement over the state-of-the-art on cross-domain WSD tasks, while exploring the strengths and weaknesses of our method.
- Applications of our sense embeddings for concept-level analyses of NLMs.

[1]Code and data: github.com/danlou/lmms

2 Language Modelling Representations

Distributional semantic representations learned from Unsupervised Neural Language Modelling (NLM) are currently used for most NLP tasks. In this section we cover aspects of word and contextual embeddings, learned from from NLMs, that are particularly relevant for our work.

2.1 Static Word Embeddings

Word embeddings are distributional semantic representations usually learned from NLM under one of two possible objectives: predict context words given a target word (Skip-Gram), or the inverse (CBOW) (word2vec, Mikolov et al., 2013). In both cases, context corresponds to a fixed-length window sliding over tokenized text, with the target word at the center. These modelling objectives are enough to produce dense vector-based representations of words that are widely used as powerful initializations on neural modelling architectures for NLP. As we explained in the introduction, word embeddings are limited by meaning conflation around word types, and reduce NLM to fixed representations that are insensitive to contexts. However, with fastText (Bojanowski et al., 2017) we're not restricted to a finite set of representations and can compositionally derive representations for word types unseen during training.

2.2 Contextual Embeddings

The key differentiation of contextual embeddings is that they are context-sensitive, allowing the same word types to be represented differently according to the contexts in which they occurr. In order to be able to produce new representations induced by different contexts, contextual embeddings employ the pretrained NLM for inferences. Also, the NLM objective for contextual embeddings is usually directional, predicting the previous and/or next tokens in arbitrarily long contexts (usually sentences). ELMo (Peters et al., 2018) was the first implementation of contextual embeddings to gain wide adoption, but it was shortly after followed by BERT (Devlin et al., 2019) which achieved new state-of-art results on 11 NLP tasks. Interestingly, BERT's impressive results were obtained from task-specific fine-tuning of pretrained NLMs, instead of using them as features in more complex models, emphasizing the quality of these representations.

3 Word Sense Disambiguation (WSD)

There are several lines of research exploring different approaches for WSD (Navigli, 2009). Supervised methods have traditionally performed best, though this distinction is becoming increasingly blurred as works in supervised WSD start exploiting resources used by knowledge-based approaches (e.g. Luo et al., 2018a; Vial et al., 2018). We relate our work to the best-performing WSD methods, regardless of approach, as well as methods that may not perform as well but involve producing sense embeddings. In this section we introduce the components and related works that are most relevant for our approach.

3.1 Sense Inventory, Attributes and Relations

The most popular sense inventory is WordNet, a semantic network of general domain concepts linked by a few relations, such as synonymy and hypernymy. WordNet is organized at different abstraction levels, which we describe below. Following the notation used in related works, we represent the main structure of WordNet, called synset, with $lemma^{\#}_{POS}$, where $lemma$ corresponds to the canonical form of a word, $_{POS}$ corresponds to the sense's part-of-speech (noun, verb, adjective or adverb), and $^{\#}$ further specifies this entry.

- Synsets: groups of synonymous words that correspond to the same sense, e.g. dog^1_n.
- Lemmas: canonical forms of words, may belong to multiple synsets, e.g. *dog* is a lemma for dog^1_n and $chase^1_v$, among others.
- Senses: lemmas specifed by sense (i.e. sensekeys), e.g. *dog%1:05:00::*, and *domestic_dog%1:05:00::* are senses of dog^1_n.

Each synset has a number of attributes, of which the most relevant for this work are:

- Glosses: dictionary definitions, e.g. dog^1_n has the definition 'a member of the genus Ca...'.
- Hypernyms: 'type of' relations between synsets, e.g. dog^1_n is a hypernym of pug^1_n.
- Lexnames: syntactical and logical groupings, e.g. the lexname for dog^1_n is *noun.animal*.

In this work we're using WordNet 3.0, which contains 117,659 synsets, 206,949 unique senses, 147,306 lemmas, and 45 lexnames.

3.2 WSD State-of-the-Art

While non-distributional methods, such as Zhong and Ng (2010)'s IMS, still perform competitively, there are have been several noteworthy advancements in the last decade using distributional representations from NLMs. Iacobacci et al. (2016) improved on IMS's performance by introducing word embeddings as additional features.

Yuan et al. (2016) achieved significantly improved results by leveraging massive corpora to train a NLM based on an LSTM architecture. This work is contemporaneous with Melamud et al. (2016), and also uses a very similar approach for generating sense embeddings and relying on k-NN w/MFS for predictions. Although most performance gains stemmed from their powerful NLM, they also introduced a label propagation method that further improved results in some cases. Curiously, the objective Yuan et al. (2016) used for NLM (predicting held-out words) is very evocative of the cloze-style Masked Language Model introduced by Devlin et al. (2019). Le et al. (2018) replicated this work and offers additional insights.

Raganato et al. (2017b) trained neural sequencing models for end-to-end WSD. This work reframes WSD as a translation task where sequences of words are translated into sequences of senses. The best result was obtained with a BiLSTM trained with auxilliary losses specific to parts-of-speech and lexnames. Despite the sophisticated modelling architecture, it still performed on par with Iacobacci et al. (2016).

The works of Melamud et al. (2016) and Peters et al. (2018) using contextual embeddings for WSD showed the potential of these representations, but still performed comparably to IMS.

Addressing the issue of scarce annotations, recent works have proposed methods for using resources from knowledge-based approaches. Luo et al. (2018a) and Luo et al. (2018b) combine information from glosses present in WordNet, with NLMs based on BiLSTMs, through memory networks and co-attention mechanisms, respectively. Vial et al. (2018) follows Raganato et al. (2017b)'s BiLSTM method, but leverages the semantic network to strategically reduce the set of senses required for disambiguating words.

All of these works rely on MFS fallback. Additionally, to our knowledge, all also perform disambiguation only against the set of admissible senses given the word's lemma and part-of-speech.

3.3 Other methods with Sense Embeddings

Some works may no longer be competitive with the state-of-the-art, but nevertheless remain relevant for the development of sense embeddings. We recommend the recent survey of Camacho-Collados and Pilehvar (2018) for a thorough overview of this topic, and highlight a few of the most relevant methods. Chen et al. (2014) initializes sense embeddings using glosses and adapts the Skip-Gram objective of word2vec to learn and improve sense embeddings jointly with word embeddings. Rothe and Schütze (2015)'s AutoExtend method uses pretrained word2vec embeddings to compose sense embeddings from sets of synonymous words. Camacho-Collados et al. (2016) creates the NASARI sense embeddings using structural knowledge from large multilingual semantic networks.

These methods represent sense embeddings in the same space as the pretrained word embeddings, however, being based on fixed embedding spaces, they are much more limited in their ability to generate contextual representations to match against. Furthermore, none of these methods (or those in §3.2) achieve full-coverage of the +200K senses in WordNet.

4 Method

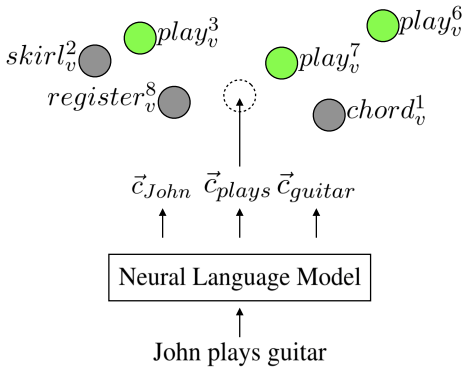

Figure 1: Illustration of our k-NN approach for WSD, which relies on full-coverage sense embeddings represented in the same space as contextualized embeddings. For simplification, we label senses as synsets. Grey nodes belong to different lemmas (see §5.3).

Our WSD approach is strictly based on k-NN (see Figure 1), unlike any of the works referred previously. We avoid relying on MFS for lemmas that do not occur in annotated corpora by generating sense embeddings with full-coverage of WordNet. Our method starts by generating sense embeddings from annotations, as done by other works, and then introduces several enhancements towards full-coverage, better performance and increased robustness. In this section, we cover each of these techniques.

4.1 Embeddings from Annotations

Our set of full-coverage sense embeddings is bootstrapped from sense-annotated corpora. Sentences containing sense-annotated tokens (or spans) are processed by a NLM in order to obtain contextual embeddings for those tokens. After collecting all sense-labeled contextual embeddings, each sense embedding is determined by averaging its corresponding contextual embeddings. Formally, given n contextual embeddings $\vec{c}$ for some sense s:

$$\vec{v}_s = \frac{1}{n}\sum_{i=1}^{n}\vec{c}_i, dim(\vec{v}_s) = 1024$$

In this work we use pretrained ELMo and BERT models to generate contextual embeddings. These models can be identified and replicated with the following details:

- ELMo: 1024 (2x512) embedding dimensions, 93.6M parameters. Embeddings from top layer (2).
- BERT: 1024 embedding dimensions, 340M parameters, cased. Embeddings from sum of top 4 layers ([-1,-4])[2].

BERT uses WordPiece tokenization that doesn't always map to token-level annotations (e.g. 'multiplication' becomes 'multi', '##plication'). We use the average of subtoken embeddings as the token-level embedding. Unless specified otherwise, our LMMS method uses BERT.

4.2 Extending Annotation Coverage

As many have emphasized before (Navigli, 2009; Camacho-Collados and Pilehvar, 2018; Le et al., 2018), the lack of sense annotations is a major limitation of supervised approaches for WSD. We address this issue by taking advantage of the semantic relations in WordNet to extend the annotated signal to other senses. Semantic networks are often explored by knowledge-based approaches, and some recent works in supervised approaches as well (Luo et al., 2018a; Vial et al., 2018). The

[2]This was the configuration that performed best out of the ones on Table 7 of Devlin et al. (2018).

guiding principle behind these approaches is that sense-level representations can be imputed (or improved) from other representations that are known to correspond to generalizations due to the network's taxonomical structure. Vial et al. (2018) leverages relations in WordNet to reduce the sense inventory to a minimal set of entries, making the task easier to model while maintaining the ability to distinguish senses. We take the inverse path of leveraging relations to produce representations for additional senses.

On §3.1 we covered synsets, hypernyms and lexnames, which correspond to increasingly abstract generalizations. Missing sense embeddings are imputed from the aggregation of sense embeddings at each of these abstraction levels. In order to get embeddings that are representative of higher-level abstractions, we simply average the embeddings of all lower-level constituents. Thus, a synset embedding corresponds to the average of all of its sense embeddings, a hypernym embedding corresponds to the average of all of its synset embeddings, and a lexname embedding corresponds to the average of a larger set of synset embeddings. All lower abstraction representations are created before next-level abstractions to ensure that higher abstractions make use of lower generalizations. More formally, given all missing senses in WordNet $\hat{s} \in W$, their synset-specific sense embeddings $S_{\hat{s}}$, hypernym-specific synset embeddings $H_{\hat{s}}$, and lexname-specific synset embeddings $L_{\hat{s}}$, the procedure has the following stages:

$$(1) \quad if|S_{\hat{s}}| > 0, \qquad \vec{v}_{\hat{s}} = \frac{1}{|S_{\hat{s}}|} \sum \vec{v}_s, \forall \vec{v}_s \in S_{\hat{s}}$$

$$(2) \quad if|H_{\hat{s}}| > 0, \quad \vec{v}_{\hat{s}} = \frac{1}{|H_{\hat{s}}|} \sum \vec{v}_{syn}, \forall \vec{v}_{syn} \in H_{\hat{s}}$$

$$(3) \quad if|L_{\hat{s}}| > 0, \quad \vec{v}_{\hat{s}} = \frac{1}{|L_{\hat{s}}|} \sum \vec{v}_{syn}, \forall \vec{v}_{syn} \in L_{\hat{s}}$$

In Table 1 we show how much coverage extends while improving both recall and precision.

		F1 / P / R (without MFS)	
Source	**Coverage**	**BERT**	**ELMo**
SemCor	16.11%	68.9 / 72.4 / 65.7	63.0 / 66.2 / 60.1
+ synset	26.97%	70.0 / 72.6 / 70.0	63.9 / 66.3 / 61.7
+ hypernym	74.70%	73.0 / 73.6 / 72.4	67.2 / 67.7 / 66.6
+ lexname	100%	73.8 / 73.8 / 73.8	68.1 / 68.1 / 68.1

Table 1: Coverage of WordNet when extending to increasingly abstract representations along with performance on the ALL test set of Raganato et al. (2017a).

4.3 Improving Senses using the Dictionary

There's a long tradition of using glosses for WSD, perhaps starting with the popular work of Lesk (1986), which has since been adapted to use distributional representations (Basile et al., 2014). As a sequence of words, the information contained in glosses can be easily represented in semantic spaces through approaches used for generating sentence embeddings. There are many methods for generating sentence embeddings, but it's been shown that a simple weighted average of word embeddings performs well (Arora et al., 2017).

Our contextual embeddings are produced from NLMs using attention mechanisms, assigning more importance to some tokens over others, so they already come 'pre-weighted' and we embed glosses simply as the average of all of their contextual embeddings (without preprocessing). We've also found that introducing synset lemmas alongside the words in the gloss helps induce better contextualized embeddings (specially when glosses are short). Finally, we make our dictionary embeddings ($\vec{v}_d$) sense-specific, rather than synset-specific, by repeating the lemma that's specific to the sense, alongside the synset's lemmas and gloss words. The result is a sense-level embedding, determined without annotations, that is represented in the same space as the sense embeddings we described in the previous section, and can be trivially combined through concatenation or average for improved performance (see Table 2).

Our empirical results show improved performance by concatenation, which we attribute to preserving complementary information from glosses. Both averaging and concatenating representations (previously L_2 normalized) also serves to smooth possible biases that may have been learned from the SemCor annotations. Note that while concatenation effectively doubles the size of our embeddings, this doesn't equal doubling the expressiveness of the distributional space, since they're two representations from the same NLM. This property also allows us to make predictions for contextual embeddings (from the same NLM) by simply repeating those embeddings twice, aligning contextual features against sense and dictionary features when computing cosine similarity. Thus, our sense embeddings become:

$$\vec{v}_s = \begin{bmatrix} ||\vec{v}_s||_2 \\ ||\vec{v}_d||_2 \end{bmatrix}, dim(\vec{v}_s) = 2048$$

Configurations	LMMS_{1024}			LMMS_{2048}			LMMS_{2348}
Embeddings							
Contextual (d=1024)	✗		✗	✗	✗		✗
Dictionary (d=1024)		✗	✗	✗		✗	✗
Static (d=300)					✗	✗	✗
Operation							
Average			✗				
Concatenation				✗	✗	✗	✗
Perf. (F1 on ALL)							
Lemma & POS	73.8	58.7	75.0	**75.4**	73.9	58.7	**75.4**
Token (Uninformed)	42.7	6.1	36.5	35.1	64.4	45.0	**66.0**

Table 2: Overview of the different performance of various setups regarding choice of embeddings and combination strategy. All results are for the 1-NN approach on the ALL test set of Raganato et al. (2017a). We also show results that ignore the lemma and part-of-speech features of the test sets to show that the inclusion of static embeddings makes the method significantly more robust to real-world scenarios where such gold features may not be available.

4.4 Morphological Robustness

WSD is expected to be performed only against the set of candidate senses that are specific to a target word's lemma. However, as we'll explain in §5.3, there are cases where it's undesirable to restrict the WSD process.

We leverage word embeddings specialized for morphological representations to make our sense embeddings more resilient to the absence of lemma features, achieving increased robustness. This addresses a problem arising from the susceptibility of contextual embeddings to become entirely detached from the morphology of their corresponding tokens, due to interactions with other tokens in the sentence.

We choose fastText (Bojanowski et al., 2017) embeddings (pretrained on CommonCrawl), which are biased towards morphology, and avoid Out-of-Vocabulary issues as explained in §2.1. We use fastText to generate static word embeddings for the lemmas ($\vec{v}_l$) corresponding to all senses, and concatenate these word embeddings to our previous embeddings. When making predictions, we also compute fastText embeddings for tokens, allowing for the same alignment explained in the previous section. This technique effectively makes sense embeddings of morphologically related lemmas more similar. Empirical results (see Table 2) show that introducing these static embeddings is crucial for achieving satisfactory performance when not filtering candidate senses.

Our final, most robust, sense embeddings are thus:

$$\vec{v}_s = \begin{bmatrix} ||\vec{v}_s||_2 \\ ||\vec{v}_d||_2 \\ ||\vec{v}_l||_2 \end{bmatrix}, dim(\vec{v}_s) = 2348$$

5 Experiments

Our experiments centered on evaluating our solution on Raganato et al. (2017a)'s set of cross-domain WSD tasks. In this section we compare our results to the current state-of-the-art, and provide results for our solution when disambiguating against the full set of possible senses in WordNet, revealing shortcomings to be improved.

5.1 All-Words Disambiguation

In Table 3 we show our results for all tasks of Raganato et al. (2017a)'s evaluation framework. We used the framework's scoring scripts to avoid any discrepancies in the scoring methodology. Note that the k-NN referred in Table 3 always refers to the closest neighbor, and relies on MFS fallbacks.

The first noteworthy result we obtained was that simply replicating Peters et al. (2018)'s method for WSD using BERT instead of ELMo, we were able to significantly, and consistently, surpass the performance of all previous works. When using our method (LMMS), performance still improves significantly over the previous impressive results (+1.9 F1 on ALL, +3.4 F1 on SemEval 2013). Interestingly, we found that our method using ELMo embeddings didn't outperform ELMo k-NN with MFS fallback, suggesting that it's necessary to achieve a minimum competence level of embeddings from sense annotations (and glosses) before the inferred sense embeddings become more useful than MFS.

In Figure 2 we show results when considering additional neighbors as valid predictions, together with a random baseline considering that some target words may have less senses than the number of accepted neighbors (always correct).

Model	Senseval2 (n=2,282)	Senseval3 (n=1,850)	SemEval2007 (n=455)	SemEval2013 (n=1,644)	SemEval2015 (n=1,022)	ALL (n=7,253)
MFS† (Most Frequent Sense)	65.6	66.0	54.5	63.8	67.1	64.8
IMS† (2010)	70.9	69.3	61.3	65.3	69.5	68.4
IMS + embeddings† (2016)	72.2	70.4	62.6	65.9	71.5	69.6
context2vec k-NN† (2016)	71.8	69.1	61.3	65.6	71.9	69.0
word2vec k-NN (2016)	67.8	62.1	58.5	66.1	66.7	-
LSTM-LP (Label Prop.) (2016)	73.8	71.8	63.5	69.5	72.6	-
Seq2Seq (Task Modelling) (2017b)	70.1	68.5	63.1*	66.5	69.2	68.6*
BiLSTM (Task Modelling) (2017b)	72.0	69.1	64.8*	66.9	71.5	69.9*
ELMo k-NN (2018)	71.5	67.5	57.1	65.3	69.9	67.9
HCAN (Hier. Co-Attention) (2018a)	72.8	70.3	-*	68.5	72.8	-*
BiLSTM w/Vocab. Reduction (2018)	72.6	70.4	61.5	70.8	71.3	70.8
BERT k-NN	**76.3**	73.2	66.2	71.7	74.1	73.5
LMMS_{2348} (ELMo)	68.1	64.7	53.8	66.9	69.0	66.2
LMMS_{2348} (BERT)	**76.3**	**75.6**	**68.1**	**75.1**	**77.0**	**75.4**

Table 3: Comparison with other works on the test sets of Raganato et al. (2017a). All works used sense annotations from SemCor as supervision, although often different pretrained embeddings. † - reproduced from Raganato et al. (2017a); * - used as a development set; bold - new state-of-the-art (SOTA); underlined - previous SOTA.

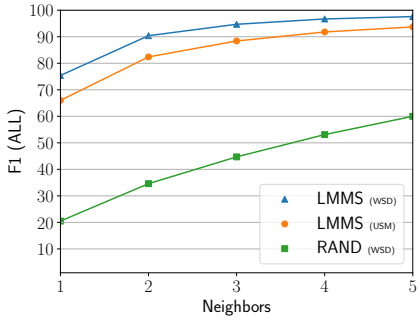

Figure 2: Performance gains with LMMS_{2348} when accepting additional neighbors as valid predictions.

5.2 Part-of-Speech Mismatches

The solution we introduced in §4.4 addressed missing lemmas, but we didn't propose a solution that addressed missing POS information. Indeed, the confusion matrix in Table 4 shows that a large number of target words corresponding to verbs are wrongly assigned senses that correspond to adjectives or nouns. We believe this result can help motivate the design of new NLM tasks that are more capable of distinguishing between verbs and non-verbs.

WN-POS	NOUN	VERB	ADJ	ADV
NOUN	96.95%	1.86%	0.86%	0.33%
VERB	9.08%	70.82%	19.98%	0.12%
ADJ	4.50%	0%	92.27%	2.93%
ADV	2.02%	0.29%	2.60%	95.09%

Table 4: POS Confusion Matrix for Uninformed Sense Matching on the ALL testset using LMMS_{2348}.

5.3 Uninformed Sense Matching

WSD tasks are usually accompanied by auxilliary parts-of-speech (POSs) and lemma features for restricting the number of possible senses to those that are specific to a given lemma and POS. Even if those features aren't provided (e.g. real-world applications), it's sensible to use lemmatizers or POS taggers to extract them for use in WSD. However, as is the case with using MFS fallbacks, this filtering step obscures the true impact of NLM representations on k-NN solutions.

Consequently, we introduce a variation on WSD, called Uninformed Sense Matching (USM), where disambiguation is always performed against the full set of sense embeddings (i.e. +200K vs. a maximum of 59). This change makes the task much harder (results on Table 2), but offers some insights into NLMs, which we cover briefly in §5.4.

5.4 Use of World Knowledge

It's well known that WSD relies on various types of knowledge, including commonsense and selectional preferences (Lenat et al., 1986; Resnik, 1997), for example. Using our sense embeddings for Uninformed Sense Matching allows us to glimpse into how NLMs may be interpreting contextual information with regards to the knowledge represented in WordNet. In Table 5 we show a few examples of senses matched at the token-level, suggesting that entities were topically understood and this information was useful to disambiguate verbs. These results would be less conclusive without full-coverage of WordNet.

Marlon*	**Brando***	**played**	**Corleone***	**in**	**Godfather***
$person_n^1$	$person_n^1$	act_v^3	$syndicate_n^1$	$movie_n^1$	$location_n^1$
$womanizer_n^1$	$group_n^1$	$make_v^{42}$	$mafia_n^1$	$telefilm_n^1$	$here_r^1$
$bustle_n^1$	$location_n^1$	$emote_v^1$	$person_n^1$	$final\_cut_n^1$	$there_r^1$

$\mathbf{act_v^3}$: play a role or part; $\mathbf{make_v^{42}}$: represent fictiously, as in a play, or pretend to be or act like; $\mathbf{emote_v^1}$: give expression or emotion to, in a stage or movie role.

Serena*	**Williams**	**played**	**Kerber***	**in**	**Wimbledon***
$person_n^1$	$professional\_tennis_n^1$	$play_v^1$	$person_n^1$	win_v^1	$tournament_n^1$
$therefore_r^1$	$tennis_n^1$	$line\_up_v^6$	$group_n^1$	$romp_v^3$	$world\_cup_n^1$
$reef_n^1$	$singles_n^1$	$curl_v^5$	$take\_orders_v^2$	$carry_v^{38}$	$elimination\_tournament_n^1$

$\mathbf{play_v^1}$: participate in games or sport; $\mathbf{line\_up_v^6}$: take one's position before a kick-off; $\mathbf{curl_v^5}$: play the Scottish game of curling.

David	**Bowie***	**played**	**Warszawa***	**in**	**Tokyo**
$person_n^1$	$person_n^1$	$play_v^{14}$	$poland_n^1$	$originate\_in_n^1$	$tokyo_n^1$
$amati_n^2$	$folk\_song_n^1$	$play_v^6$	$location_n^1$	in_r^1	$japan_n^1$
$guarnerius_n^3$	$fado_n^1$	$riff_v^2$	$here_n^1$	$take\_the\_field_v^2$	$japanese_a^1$

$\mathbf{play_v^{14}}$: perform on a certain location; $\mathbf{play_v^6}$: replay (as a melody); $\mathbf{riff_v^2}$: play riffs.

Table 5: Examples controlled for syntactical changes to show how the correct sense for 'played' can be induced accordingly with the mentioned entities, suggesting that disambiguation is supported by world knowledge learned during LM pretraining. Words with * never occurred in SemCor. Senses shown correspond to the top 3 matches in LMMS_{1024} for each token's contextual embedding (uninformed). For clarification, below each set of matches are the WordNet definitions for the top disambiguated senses of 'played'.

6 Other Applications

Analyses of conventional word embeddings have revealed gender or stereotype biases (Bolukbasi et al., 2016; Caliskan et al., 2017) that may have unintended consequences in downstream applications. With contextual embeddings we don't have sets of concept-level representations for performing similar analyses. Word representations can naturally be derived from averaging their contextual embeddings occurring in corpora, but then we're back to the meaning conflation issue described earlier. We believe that our sense embeddings can be used as representations for more easily making such analyses of NLMs. In Figure 3 we provide an example that showcases meaningful differences in gender bias, including for lemmas shared by different senses (*doctor*: PhD vs. medic, and *counselor*: therapist vs. summer camp supervisor). The bias score for a given synset s was calculated as following:

$$bias(s) = sim(\vec{v}_{man_n^1}, \vec{v}_s) - sim(\vec{v}_{woman_n^1}, \vec{v}_s)$$

Besides concept-level analyses, these sense embeddings can also be useful in applications that don't rely on a particular inventory of senses. In Loureiro and Jorge (2019), we show how similarities between matched sense embeddings and contextual embeddings are used for training a classifier that determines whether a word that occurs in two different sentences shares the same meaning.

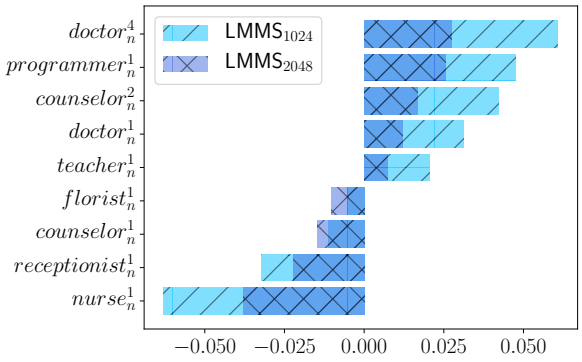

Figure 3: Examples of gender bias found in the sense vectors. Positive values quantify bias towards man_n^1, while negative values quantify bias towards $woman_n^1$.

7 Future Work

In future work we plan to use multilingual resources (i.e. embeddings and glosses) for improving our sense embeddings and evaluating on multilingual WSD. We're also considering exploring a semi-supervised approach where our best embeddings would be employed to automatically annotate corpora, and repeat the process described on this paper until convergence, iteratively fine-tuning sense embeddings. We expect our sense embeddings to be particularly useful in downstream tasks that may benefit from relational knowledge made accessible through linking words (or spans) to commonsense-level concepts in WordNet, such as Natural Language Inference.

8 Conclusion

This paper introduces a method for generating sense embeddings that allows a clear improvement of the current state-of-the-art on cross-domain WSD tasks. We leverage contextual embeddings, semantic networks and glosses to achieve full-coverage of all WordNet senses. Consequently, we're able to perform WSD with a simple 1-NN, without recourse to MFS fallbacks or task-specific modelling. Furthermore, we introduce a variant on WSD for matching contextual embeddings to all WordNet senses, offering a better understanding of the strengths and weaknesses of representations from NLM. Finally, we explore applications of our sense embeddings beyond WSD, such as gender bias analyses.

9 Acknowledgements

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project: UID/EEA/50014/2019.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4356–4364, USA. Curran Associates Inc.

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

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, 63(1):743–788.

Jose Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36 – 64.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805v1.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christiane Fellbaum. 1998. In *WordNet : an electronic lexical database*. MIT Press.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Minh Le, Marten Postma, Jacopo Urbani, and Piek Vossen. 2018. A deep dive into word sense disambiguation with LSTM. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 354–365, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Doug Lenat, Mayank Prakash, and Mary Shepherd. 1986. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquistion bottlenecks. *AI Mag.*, 6(4):65–85.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Daniel Loureiro and Alípio Mário Jorge. 2019. Liaad at semdeep-5 challenge: Word-in-context (wic). In *SemDeep-5@IJCAI 2019*, page forthcoming.

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411, Brussels, Belgium. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. 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, pages 3111–3119, USA. Curran Associates Inc.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. 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: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *Tagging Text with Lexical Semantics: Why, What, and How?*

Sascha Rothe and Hinrich Schütze. 2015. 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 (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2018. Improving the coverage and the generalization ability of neural word sense disambiguation through hypernymy and hyponymy relationships. *CoRR*, abs/1811.00960.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.