Breaking Through the 80% Glass Ceiling:
Raising the State of the Art in Word Sense Disambiguation
by Incorporating Knowledge Graph Information

Michele Bevilacqua and **Roberto Navigli**
Sapienza NLP Group
Department of Computer Science
Sapienza University of Rome
`{bevilacqua,navigli}@di.uniroma1.it`

Abstract

Neural architectures are the current state of the art in Word Sense Disambiguation (WSD). However, they make limited use of the vast amount of relational information encoded in Lexical Knowledge Bases (LKB). We present *Enhanced WSD Integrating Synset Embeddings and Relations* (EWISER), a neural supervised architecture that is able to tap into this wealth of knowledge by embedding information from the LKB graph within the neural architecture, and to exploit pretrained synset embeddings, enabling the network to predict synsets that are not in the training set. As a result, we set a new state of the art on almost all the evaluation settings considered, also breaking through, for the first time, the 80% ceiling on the concatenation of all the standard all-words English WSD evaluation benchmarks. On multilingual all-words WSD, we report state-of-the-art results by training on nothing but English.

1 Introduction

There is a growing body of research dealing with the integration of prior knowledge into neural networks for Natural Language Processing (NLP) tasks, be it through pretraining on self-supervised tasks such as language modeling (Peters et al., 2018; Devlin et al., 2019), or through the incorporation of information from knowledge bases (Peters et al., 2019; Logan et al., 2019). In Word Sense Disambiguation (WSD), i.e., the task of associating a word in context with the most appropriate meaning from a finite set of possible choices (Navigli, 2009), the gap between supervision and knowledge (Navigli, 2018) has been overcome by several efforts directed at learning effective vector representations (Loureiro and Jorge, 2019; Scarlini et al., 2020) in the same space as contextualized embeddings, and exploring the usage of definitional knowledge in supervised sequence learning neural architectures (Luo et al., 2018; Kumar et al., 2019; Huang et al., 2019).

However, the Lexical Knowledge Bases (LKBs) from which such information is retrieved, such as WordNet (Miller, 1995) and BabelNet (Navigli and Ponzetto, 2012), also provide a great wealth of relational knowledge in structured form (i.e., hypernymy, meronymy, similarity, etc.), which is often neglected due to the non-trivial integration of data of this kind into neural architectures. Even though such information can, instead, be exploited by knowledge-based WSD algorithms (Agirre and Soroa, 2009; Moro et al., 2014), rivaling supervised pre-contextualized embedding approaches (Maru et al., 2019), the performances still lag behind (Huang et al., 2019; Vial et al., 2019).

Building on *Extended WSD Integrating Sense Embeddings* (EWISE) (Kumar et al., 2019), a neural WSD system incorporating prior knowledge through synset embeddings, we present *Enhanced WSD Integrating Synset Embeddings and Relations* (EWISER), a hybrid knowledge-based and supervised approach to WSD that integrates explicit relational information from the WordNet LKB. Our approach offers the following contributions:

1. We introduce the novel *structured logits* mechanism, which enables the exploitation of concept relatedness as determined by LKB edges. In our method, pre-softmax scores are a weighted combination of synset-specific scores, and can be computed via dot product with a sparse adjacency matrix.

2. We generalise the sense vector dot product technique from EWISE, showing that off-the-shelf pretrained embeddings can be used.

3. We show that the structured logits mechanism and the use of sense embeddings are orthogonal and can be exploited jointly.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2854–2864
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

Our approach is simple and extensible, does not require fine tuning of contextualized embeddings, and has a very modest parameter budget apart from synset embeddings. EWISER achieves a new state of the art in all-words English WSD. Moreover, we obtain state-of-the-art performances on the cross-lingual all-words WSD evaluation, without using non-English training data.

2 Related Work

Supervised WSD Supervised systems have to rely on expensive hand-labeled data to achieve good results (Pasini, 2020). The best approaches currently rely on neural networks. The model presented by Raganato et al. (2017) formulates the task as a token classification problem, with an LSTM with attention classifier producing a probability distribution over both words and senses. Subsequent work has shown that better results can be obtained by only having scores for senses or synsets (Vial et al., 2019). Shallower, simpler networks can achieve even better performances (Uslu et al., 2018).

Contextualized vectors can be exploited in token tagging architectures (Vial et al., 2019; Bevilacqua and Navigli, 2019; Hadiwinoto et al., 2019). However, purely supervised systems are dependent on the data they are trained on, therefore when some sense is underrepresented in the training corpus it is not easy for them to predict it.

LKBs in Supervised WSD More closely related to the core of our contribution, LKB information, such as natural language definitions of word meaning, can be exploited in neural token tagging architectures. For example, in GlossBERT (Huang et al., 2019) a pretrained BERT encoder is fed both the context sentence and the gloss, and is trained to predict whether the gloss correctly describes the use of the target word. Successful results have been obtained by encoding glosses in dense vectors (Luo et al., 2018).

In EWISE (Kumar et al., 2019), WSD is performed in a two-step process: first, gloss embeddings are produced through a training procedure that also takes into account the WordNet's graph structure; then, the gloss embeddings are scored via dot product with a contextual vector computed with an LSTM model, which is trained through regular categorical cross-entropy. Our work builds on top of EWISE in that it generalizes its sense vector dot product approach, but features a novel mechanism that injects relational knowledge into the architecture through a simple additional sparse dot product operation. Moreover, we show that better performances can be obtained by training the output embedding matrix, and that different sense/synset vectors can be used to initialize the output embeddings.

Note that our approach is different from that of Vial et al. (2019), in that we do not conflate senses together through the use of WordNet hypernymy; rather, we mantain all the original meaning distinctions, and exploit the logit scores over the full vocabulary in a second, distinct step.

3 EWISER: Neural WSD with More Prior Knowledge

3.1 WSD as a classification problem

WSD can be treated as a simple token classification problem, similar to POS tagging or Named Entity Recognition. As such, abstracting away from all the intricacies of any particular supervised model, we need to produce a vector representation $\mathbf{h} \in \mathbb{R}^d$ of a target word in a given context, and use it to yield a probability distribution over all its possible labels, i.e., its senses or synsets. The simplest way to do this is to learn a weight matrix $O \in \mathbb{R}^{d \times |\mathcal{V}|}$, where $\mathcal{V}$ is the output vocabulary[1], and compute a vector of unnormalized scores $\mathbf{z}$ as the product of $\mathbf{h}^T$ and O. Having multiple instances to classify packed into the matrix H, we can compute all the scores at the same time by a single dot product followed by a sum over columns with a bias vector:

$$Z = HO + \mathbf{b} \tag{1}$$

Finally, Z is transformed into a probability distribution through a standard softmax activation function. Typically, O is randomly initialized, and just trained end-to-end with the rest of the architecture (Raganato et al., 2017; Vial et al., 2019; Bevilacqua and Navigli, 2019). During training the categorical cross-entropy loss is computed for each instance Z_i. At inference time, the model predicts the synset $\hat{s}$ with the highest probability among the set $S(w_i) \subset \mathcal{V}$ of possible synsets for word w_i:

$$\hat{s}_i = \operatorname*{argmax}_{s \in S(w_i)} Z_{i,s} \tag{2}$$

where, for each w_i, $S(w_i)$ depends on both the lemma and its part-of-speech, and is determined by the WordNet inventory.

[1] We use synsets as output vocabulary.

3.2 Neural WSD Architecture

We now describe a simple neural WSD architecture to be used as the core on top of which we will integrate the EWISER additions. For each word to disambiguate, our network takes as input the sum of the outputs of the last 4 layers of BERT Large (cased) and uses a 2-layer feedforward to compute the logit scores Z:

$$\begin{aligned} B &= B_{-4} + B_{-3} + B_{-2} + B_{-1} \\ H_0 &= \mathrm{BatchNorm}(B) \\ H_1 &= \mathrm{swish}(H_0 W + \mathbf{b}) \\ Z &= H_1 O \end{aligned} \tag{3}$$

where W, $\mathbf{b}$ are parameters of the models, and B_{-4} to B_{-1} are BERT hidden states[2]. We employ the swish activation function (Ramachandran et al., 2018), which has shown very promising results in NLP (Eger et al., 2018).

Note that, while our architecture is very simple, it would be straightforward to incorporate powerful additions such as a sequence encoder – like an LSTM or a Transformer (Vaswani et al., 2017) classifier. While this might indeed produce better performances, improvements of this kind are not directly pertinent to our contribution.

3.3 Structured Logits

The matrix multiplication in Equation 1 is wasteful during both training and inference, as it produces scores over the entire vocabulary $\mathcal{V}$, even though the number of possible synsets is much smaller than the cardinality of $\mathcal{V}$. Since the model is equally penalized by the cross-entropy loss when it gives a high score to a synset either related or unrelated to the correct one, there is little incentive to learn similar vectors for related synsets. Moreover, computing logits over the whole vocabulary does not bring any benefit in inference, as each score is computed independently, without taking into account connections between output classes.

We address this issue by devising an architecture, i.e., EWISER, that can inject into the network relatedness knowledge as encoded in an arbitrary graph, and use it in training as well as in inference.

3.3.1 Synset Graph in EWISER

As LKBs are structured into graphs, we want to be able to exploit, when computing the probability distribution vector over $\mathcal{V}$ for a target word, the explicit information of an arbitrary weighted graph $G = \langle V, E, w\rangle$, where $w : E \to \mathbb{R}$, and the vertices $V = \mathcal{V}$ – i.e., the nodes are synsets. Instead of using the vector $\mathbf{z}$ for prediction, we compute another vector $\mathbf{q}$ where for each component, i.e.. for each synset s, the score synset $\mathbf{q}_s$ is a function of both the "hidden" score $\mathbf{z}_s$ for s, and the hidden scores $\mathbf{z}_{s'}$ for all synsets s' such that there is an edge $\langle s', s\rangle \in E$. In order to do this, we calculate $\mathbf{q}_s$ as $\mathbf{z}_s$ plus the sum of the products of $\mathbf{z}'_s$ and the weight of the edge $\langle s', s\rangle$.

$$\mathbf{q}_s = \mathbf{z}_s + \sum_{s' \in V | \langle s', s\rangle \in E} w(\langle s', s\rangle) \cdot \mathbf{z}_{s'} \tag{4}$$

As a result, $\mathbf{q}_s$ is a weighted combination of the scores for all the output vocabulary. In Figure 1 we show this process visually.

3.3.2 Computing Q

The most natural way to encode the graph G is with the adjacency matrix A, in which $A_{s_1 s_2} = w(\langle s_1, s_2\rangle)$. If $A_{s_1 s_2} = 0$ there is no edge between the two synsets. The new logits matrix Q can be obtained efficiently by simply computing the dot product between the hidden logits Z and the transposed adjacency matrix A^T, summing Z to the results.

$$\begin{aligned} Z &= HO + \mathbf{b} \\ Q &= ZA^T + Z \end{aligned} \tag{5}$$

Finally, we apply the softmax function to Q to get the probabilities.

3.3.3 The matrix A

In our case, we build the graph and adjacency matrix A from the relations between synsets or senses in WordNet. As WordNet relations are not weighted, for every synset s we set $A_{s',s}$ to $1/N$, where N is the number of incoming connections. In this way we avoid imbalanced predictions towards synsets with more incoming connections.

We experiment with including different relations in A. Our base configuration includes *similarity*, *verb group*, and *derivationally related*[3] edges. As for *hypernymy* and its inverse, *hyponymy*, we experiment with different possible ways of including them in A: (i) including only hypernymy (**hyper**); (ii) only hyponymy (**hypo**); (iii) both hypernymy

[2] If a token consists of more than one subword, we average its subword representations.

[3] We connect two synsets with a *derivationally related* edge if at least one pair of senses therein is connected via a *derivationally related* edge.

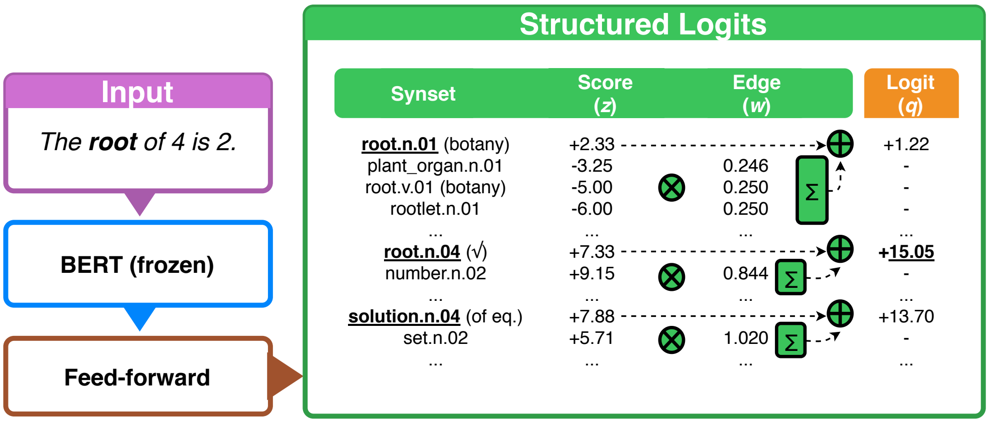
Input
The root of 4 is 2.
BERT (frozen)
Feed-forward
Structured Logits
Synset
Score (z)
Edge (w)
Logit (q)
root.n.01 (botany) +2.33 +1.22
plant_organ.n.01 -3.25 0.246 -
root.v.01 (botany) -5.00 0.250 -
rootlet.n.01 -6.00 0.250 -
root.n.04 (√) +7.33 +15.05
number.n.02 +9.15 0.844 -
solution.n.04 (of eq.) +7.88 +13.70
set.n.02 +5.71 1.020 -
Σ
...

Figure 1: The structured logits mechanism in EWISER. The example input is the sentence "The *root* of 4 is 2." Scores for a selection of synsets representing possible senses of *root* are shown. Going from left to right, the "hidden" logits ($\mathbf{z}$) of related synsets are multiplied by the edge weights, summed together, and then added to the "hidden" logits of the related synsets, resulting in the "final" logits ($\mathbf{q}$).

and hyponymy (**hyper+hypo**); (iv) the transitive closure over hypernymy (the set of relations that are obtained by following hypernymy paths) (**hyper***); (v) the transitive closure over hypernymy and hyponymy (**hyper+hypo***);

Informally, hypernymy and hyponymy correspond to different kinds of reasoning, which might be characterized as, respectively, *inductive* ("if it is an electronic device, then it might be a mouse") and *deductive* ("if it is a mouse, then it is an electronic device"). The closures are a way to flatten the hierarchy, thus enabling multi-hop reasoning by making the $\mathbf{q}_s$ score dependent on the $\mathbf{z}$ scores for synsets whose path distance to s is greater than 1 in the original graph.

Fine-tuning the adjacency matrix If weights in A are frozen, every connected synset gives an equal contribution to the final score $\mathbf{q}_s$. However, it is also reasonable to assume that not all synsets are equally relevant. For example, the score for *inanimate object* should be less relevant than that for *device* for predicting the hardware meaning of *mouse*. Thus, we experiment on fine-tuning A by only updating non-zero weights.

3.4 Output Layer Weights

While O can be seen as just the final linear map in the network, it is also reasonable to think about it as a counterpart of an embedding matrix. Whereas in the intermediate layers of the neural network there is no one-to-one mapping between values of the matrix and input or output classes, in O there is a distinct column for each of the elements in $\mathcal{V}$. As a matter of fact, the logit of synset s ($\mathbf{z}_s$) is just the scalar product between $\mathbf{h}$ and O_s^T, i.e., the column in O associated with s. So, just as with word embeddings, O can be seen as a collection for vector representations that have one-to-one mappings to output classes. Thus, it is possible to use synset embeddings to provide a better initialization for O than random. This idea has already been exploited by EWISE (Kumar et al., 2019), in which logit scores over $\mathcal{V}$ are computed by dot product between the hidden vector $\mathbf{h}$ and the gloss embedding vector $\mathbf{g}^{(\mathbf{s})}$ as follows:

$$\mathbf{z}_s = \mathbf{h}^T\mathbf{g}^{(\mathbf{s})} + \mathbf{b}^T\mathbf{g}^{(\mathbf{s})} \qquad (6)$$

where $\mathbf{b}$ is a learned bias vector. Note that if we pack the synset gloss vector $\mathbf{g}^{(\mathbf{s})}$ for every $s \in \mathcal{V}$ into the O matrix, this looks almost identical to the canonical linear layer in Eq. 1, with the only difference being the fact that the bias is now the result of the dot product between $\mathbf{b}$ and O, rather than being directly parametrized as a vector $\in \mathbb{R}^{|\mathcal{V}|}$.

3.4.1 Weight Training vs. Freezing vs. Thawing

In EWISE, the sense embeddings are learned independently from the WSD system and kept frozen during training. It is worth exploring whether better results can be achieved by allowing further refining of the weights during training. We expect initialization and freezing (which we refer to as, respectively, O-init and O-freeze) to have different effects depending on whether the gold synset is found in the training set. If weights are initialized and then updated during training, the columns in O corresponding to unattested synsets will only receive a "negative" signal from the cross-entropy loss; conversely, attested synsets can be further refined and predicted more accurately. If weights are frozen, the architecture will have to accommodate to the pretrained synset representations, meaning that, especially if there is no learned bias, it will be easier to predict unseen classes. No fine-tuning may, however, result in diminished performance, as the pre-trained synset representations are not tailored to WSD. An additional possibility to achieve better transfer between the information in the embeddings and the WSD system is to use a freeze-then-thaw scheme, similar to the chain-thaw method of Howard and Ruder (2018). The approach entails training an O-freeze model, restoring the best checkpoint, and then doing further training with O "thawed", i.e., with trainable weights.

4 Experiments

We assess the performance of EWISER in all-words English WSD, against both a simple but competitive baseline, i.e., the simple feedforward network taking BERT hidden states as input described in Section 3.2, and state-of-art approaches. We first experiment separately on the integration of explicit relational information through structured logits (Section 4.1), and the integration of synset embeddings through the initialization of O (Section 4.2). Then, building on the results of these experiments, we evaluate the full EWISER architecture (Section 4.3). Finally, we assess our approach on cross-lingual WSD (Section 4.4), training on English and evaluating on French, German, Italian and Spanish.

4.1 Structured Logits

As explained in Section 3.3.2, in EWISER, relational knowledge is integrated through a dot product between the logits matrix Z and the transposed adjacency matrix A^T. We perform experiments with different configurations that vary according to which edges are included in A.

4.1.1 Setting

We experiment with the edge sets which are listed in Section 3.3.3. For each configuration we evaluate two different training runs, one in which A is frozen (A-**freeze**), and the other where edge weights are trained (A-**train**). We contrast the performance of the models with the above-mentioned baseline.

Model Arch.		ALL	No15	No15$^-$
baseline	–	74.2	73.9	52.2
hyper	A-freeze	75.6	75.4	59.8
	A-train	**75.9**	**75.5**	59.2
hypo	A-freeze	74.6	74.4	57.7
	A-train	74.6	74.3	54.5
hyper+hypo	A-freeze	75.7	**75.5**	59.8
	A-train	75.7	75.4	57.7
hyper*	A-freeze	75.2	75.0	58.6
	A-train	75.4	75.3	57.7
hyper+hypo*	A-freeze	75.4	75.3	**59.9**
	A-train	74.7	74.4	56.5

Table 1: Evaluation of structured logits on English all-words WSD. F1 is reported.

4.1.2 Data & Hyperparameters

We train the baseline and the configurations under comparison on SemCor (Miller et al., 1994) for 20 epochs, with a batch size of 4000 tokens. We do not employ sentences as context. Rather, we split documents in chunks of at most 100 tokens. The hidden size of the 2-layer feedforward is 512, with a dropout value of 0.2. The optimizer is Adam (Kingma and Ba, 2015), which we employ with a learning rate of 10^{-4}. Following Bevilacqua and Navigli (2019), we select as development set (to select the best epoch) the SemEval-2015 dataset (Moro and Navigli, 2015). As customary, we report the results on the concatenation (**ALL**) of all the evaluation datasets from Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-2007 (Pradhan et al., 2007), SemEval-2013 (Navigli et al., 2013), and the aforementioned SemEval-2015. In addition, we report performances on ALL with all instances from the development set removed (**No15**), and on the subset of No15 whose gold synsets do not appear in SemCor (**No15$^-$**).

4.1.3 Results

We report in Table 1 the results of the experiments on the addition of structured logits to the baseline architecture.

As can be seen, the use of hypernyms brings the biggest gain to performances, with the strongest improvement against the baseline reported with simple hypernymy and fine-tuning of A: 1.7 points on ALL and 1.6 on No15. The closures, i.e., hyper* and hyper+hypo*, do not seem to be very beneficial, achieving slightly worse results than the simple counterpart. Much of the improvement seems to come from the increased performance of the unseen split No15$^-$ where the gold is not in SemCor, with an absolute improvement of 7.6 points with hypernymy edges and no fine-tuning, and of 7 points with hypernymy edges and fine-tuning. Fine-tuning A makes for better results than keeping the weights of the adjacency matrix fixed on both ALL and No15, but results in slight-to-moderate decreases on No15$^-$, as the network is able to adjust the weights in order to bring down the $\mathbf{q}$ scores for unseen synsets.

4.2 Output Embeddings

As in EWISE, in EWISER logits are computed by a dot product between a matrix of hidden scores and output synset embeddings. However, we do not train our own synset embeddings: rather, we employ off-the-shelf vectors. In this section we evaluate the performance of different options both in the choice of the embeddings and in how they are integrated into the network. We contrast the performance with our baseline, in which the O matrix is randomly initialized and the embeddings are trained.

4.2.1 Setting

We experiment with different options for the initialization of O:

Deconf 300*d* We use the 300-dimensional vectors released by Pilehvar and Collier (2016), which are built from Word2Vec Google news word embeddings.

LMMS 2048*d* We use the 2048-dimensional vectors produced by Loureiro and Jorge (2019), built as the concatenation of BERT Large cased states' centroids for instances in SemCor with the synset gloss vector, computed from BERT Large states as well. We normalize the vectors to unit length. Since LMMS vectors are quite big, we reduce the number of dimensions to 512 with truncated SVD.

SensEmBERT+LMMS 2048*d* SensEmBERT (Scarlini et al., 2020) enhances LMMS by exploiting BabelNet and Wikipedia. SensEmBERT only includes nouns, but its vectors are in the same space as LMMS, so we use the former in combination with verbs, adjectives and adverbs from the latter. We employ the same preprocessing as with LMMS.

Model Arch.		**ALL**	**No15**	**No15$^-$**
baseline	–	74.2	73.9	52.2
Deconf	O-init	75.3	75.2	55.2
	O-freeze	66.4	66.0	**72.2**
	O-thaw	75.3	75.2	60.5
	O-thaw*	73.8	73.7	62.3
LMMS	O-init	75.5	75.4	55.1
	O-freeze	75.9	75.4	59.4
	O-thaw	75.4	75.0	57.4
	O-thaw*	75.8	75.4	57.3
LMMS + SensEmBERT	O-init	76.1	76.0	59.4
	O-freeze	76.3	76.0	64.7
	O-thaw	76.4	76.1	62.3
	O-thaw*	**76.7**	**76.6**	63.4

Table 2: Evaluation of O initialization and training strategies on English all-words WSD. F1 is reported.

For each sense embedding system, we report results with four different training schemes: plain initialization (**O-init**); initialization and freezing (**O-freeze**); restore the best O-freeze, then thaw the weights of O (**O-thaw**); the same as for O-thaw, but reducing the learning rate to 10^{-5} (**O-thaw***). In all cases, synset embeddings are computed as the centroid of the senses contained in the synset.

4.2.2 Data & Hyperparameters

We train our baseline and O-init models for 20 epochs. The O-freeze model, which is much slower to converge, is trained for a maximum of 80 epochs. O-thaw and O-thaw* are trained for 10 epochs. The data on which we train and report the performances are the same as in Section 4.1.2.

4.2.3 Results

We report in Table 2 the results of the evaluation of the use of synset embeddings for the initialization of the O output embeddings matrix.

In general, the approach enables much better F1 scores compared to the baseline, but is very dependent on the quality of the embeddings, and on whether they incorporate supervision from SemCor. When using Deconf, which uses the WordNet graph to "deconflate" word-level Word2Vec vectors, with no use of training corpora, the O-freeze strategy produces the best result on No15$^-$, i.e., 72.2, with an absolute increase of 20 points over the baseline. However, O-freeze with Deconf also achieves the worst result on both ALL and No15, indicating that some form of biasing towards the most frequent synsets, which is an effect of corpus supervision, is required for the global evaluation. Fine-tuning O enables the model to obtain a decent

S	**G**	**G**$^+$	**E**	**System**	**ALL**	**No15**	**No15**$^-$	**S2**	**S3**	**S7**	**S13**	**S15**	**N**	**V**	**A**	**R**
✓	✓	-	-	Kumar et al. (2019)	71.8	70.9*	-	73.8	71.1	67.3	69.4	74.5	74.0	60.2	78.0	82.1
✓	✓	-	-	Loureiro and Jorge (2019)	75.4	75.2*	-	76.3	75.6	68.1	75.1	77.0	-	-	-	-
✓	-	-	-	Hadiwinoto et al. (2019)	73.7*	73.2*	-	75.5	73.6	68.1	71.1	76.2	-	-	-	-
✓	✓	-	-	Huang et al. (2019)	77.0$\star$	76.2*	-	77.7	75.2	72.5	76.1	**80.4**	-	-	-	-
✓	✓	-	-	Scarlini et al. (2020) - Sup.	-	-	-	-	-	-	78.7	-	80.4	-	-	-
✓	-	-	-	Vial et al. (2019)	75.6	-	-	-	-	-	-	-	-	-	-	-
✓	-	-	-	Vial et al. (2019) - ENS	76.7	76.5*	-	77.5	77.4	69.5	76.0	78.3	79.6	65.9	79.5	85.5
✓	†	-	-	EWISER$_{hyper}$	77.0$\star$	76.9	60.4	77.5	77.9	**71.0**	76.4	77.8	79.9	**66.4**	79.0	85.5
✓	✓	-	-	EWISER$_{hyper}$	77.5	77.3	68.2	78.4	77.4	**71.0**	77.4	78.7	80.7	65.1	80.9	86.1
✓	†	-	-	EWISER$_{hyper+hypo}$	76.8	76.8	59.5	77.7	77.9	70.3	76.2	76.3	79.4	65.9	80.0	**86.7**
✓	✓	-	-	EWISER$_{hyper+hypo}$	**78.3**	**78.2**	**69.1**	**78.9**	**78.4**	**71.0**	**78.9**	79.3	**81.7**	66.3	**81.2**	85.8
✓	✓	✓	✓	Vial et al. (2019)	77.1	-	-	-	-	-	-	-	-	-	-	-
✓	✓	✓	✓	Vial et al. (2019) - ENS	79.0$\star$	78.4*	-	79.7	77.8	73.4	78.7	**82.6**	81.4	68.7	**83.7**	85.5
✓	✓	✓	✓	EWISER$_{hyper}$	**80.1**	**79.8**	**75.2**	**80.8**	**79.0**	**75.2**	**80.7**	81.8	**82.9**	**69.4**	83.6	87.3
✓	✓	✓	✓	EWISER$_{hyper+hypo}$	79.8	79.3	75.1	80.2	78.5	73.8	80.6	82.3	82.7	68.5	82.9	**87.6**
-	-	-	-	Scozzafava et al. (2020)	71.7	71.0*	-	71.6	72.0	59.3	72.2	75.8	-	-	-	-
-	✓	-	-	Scarlini et al. (2020) - KB	-	-	-	-	-	-	74.8	-	75.9	-	-	-

Table 3: Evaluation of the joint use of structured logits and O-thaw* on English all-words WSD. F1 is reported. The column blocks report (i) the training corpora and system compared; (ii) overall F1; (iii) single dataset F1; (iv) POS-specific F1. †: Incorporates gloss information through synset embeddings. *: Computed from reported scores. $\star$: highest F1 that is statistically different from the best one (χ^2 with p=0.1).

F1 score, with the exception of O-thaw*, where the training run was underfitting. With LMMS, higher results are obtained, especially when freezing the weights. SensEmBERT with the LMMS backoff achieves the best results on both ALL and No15, with O-thaw* reaching at least 76.6 on ALL and No15. Probably due to the fact that SensEmBERT relies less on the supervision from SemCor, very strong results are obtained on No15$^-$ as well, with a margin of over 12 points above the baseline.

As for the training scheme adopted, the best results are obtained from the freeze-then-thaw strategy with learning rate reduction (O-thaw*) and from the simple freezing of O. Thawing consistently raises the accuracy on ALL and No15, but lowers it on No15$^-$, meaning that the fine-tuning of O shifts the balance of the trade-off between performances on seen and unseen synsets to the benefit of the former. O-init still improves over the baseline, but is less effective than its alternatives.

4.3 Combining Relational Knowledge and Sense Embeddings

Bringing everything together, we now evaluate the joint exploitation of the O initialization and structured logits in EWISER.

4.3.1 Setting

Building on the results of the previous experiments, we limit the number of model variants by only including the configurations that separately yielded the best results, namely: (i) the use of hypernyms (EWISER$_{hyper}$) or hypernyms plus hyponyms (EWISER$_{hyper+hypo}$) in the graph encoded in A, training the adjacency matrix, and (ii) the combination of SensEmBERT and LMMS for the output embeddings, trained according to the O-thaw* scheme, i.e., the freeze-then-thaw approach, with the learning rate set to 10^{-5}.

4.3.2 Data & Hyperparameters

In order to make the results of EWISER comparable to those of the state-of-the-art approaches to WSD, we report results when training not only on SemCor (**S**), but also on the union of SemCor and untagged WordNet glosses (**G**), and on the union of SemCor, tagged WordNet glosses (**G**$^+$), and WordNet examples (**E**) as well. When training on glosses, we prepend the lemma of the main sense and a semicolon to the raw gloss, and treat the added word as a tagged instance. We evaluate the model on the datasets mentioned in Section 4.1.2.

4.3.3 Results

In Table 3 we report the results of the unified evaluation. In addition to our systems, we include in the comparison the best systems from the literature, grouping the two sets together in two internally comparable blocks: (i) systems trained on SemCor, possibly making use of LKB information such as untagged glosses or the WordNet graph; (ii) systems that also make use of tagged glosses and examples; (iii) the best performing knowledge-based systems.

In almost every setting compared, EWISER outperforms the previous state of the art. Among systems in the first block (S/G) EWISER$_{hyper+hypo}$ trained on S+G obtains the best results on all the datasets except for SemEval-2015, with a margin over the two best performing systems, i.e., GlossBERT and the ensemble of 8 models of Vial et al. (2019), of, respectively, 1.3 and 1.6 points on ALL, and of 2.0 and 1.7 on No15, which does not include our dev set. Even if they do not train on untagged glosses, both EWISER$_{hyper}$ and EWISER$_{hyper+hypo}$ show comparable performances to GlossBERT on ALL, and better on No15 – without fine-tuning BERT, and with much less compute power required. The results on No15^{-}, where EWISER$_{hyper+hypo}$ with glosses achieves an F1 of 69.1, almost 10 points more than when not using them, show that definitional knowledge is beneficial for the zero-shot setting.

Adding tagged glosses and WordNet examples further boosts performances, with the best configuration, EWISER$_{hyper}$, breaking through the 80 points ceiling on ALL, an estimated upper bound on human inter-annotator agreement that is often quoted as the glass ceiling for WSD performance (Navigli, 2009). The only model we can compare with, i.e., the one of Vial et al. (2019), is outperformed on every dataset except for SemEval-2015. On ALL and No15, however, we outscore the competitor by a margin of 1.1 and 1.4 points, establishing a new state of the art in English all-words WSD. The bigger training set improves performances on No15^{-}, though the gap is not quite closed.

Not surprisingly, even the best knowledge-based systems do not offer competitive performances, since they cannot take advantage of training corpus supervision.

4.4 Cross-lingual WSD

To see whether the strong performances of EWISER carry over to the multilingual setting, we retrain the best global configuration, i.e., EWISER$_{hyper}$ trained on SemCor, WordNet's tagged glosses and usage examples, with BERT multilingual cased. We compare our system against (i) the state of the art in multilingual WSD, i.e. SensEmBERT, which can, however, only disambiguate nouns; (ii) the best performing all-PoS system, i.e. SyntagRank (Scozzafava et al., 2020), a knowledge-based system; (iii) the feedforward baseline. We report results on the French, German, Italian and Spanish all-words evaluation datasets from SemEval-2013, which contain only nouns, and the Italian and Spanish datasets from SemEval-2015, which contain all PoS. We use the revised version of the evaluation datasets[4], which is updated to be consistent with the 4.0.1 release of the BabelNet graph. As a result, we can test on a larger number of instances than previously possible.

	S13				S15	
	DE	**ES**	**FR**	**IT**	**ES**	**IT**
Scozzafava et al. (2020)	76.4	74.1	70.3	72.1	63.4	69.0
Scarlini et al. (2020)	79.2*	73.4*	77.8*	69.8*	-	-
Ours (baseline)	**81.7**	76.6	80.8	77.2	67.3	70.6
Ours (EWISER)	80.9	**78.8**	**83.6**	**77.7**	**69.5**	**71.8**

Table 4: Evaluation of the joint use of structured logits and O-thaw* on cross-lingual WSD. F1 is reported. *: Recomputed by the authors.

We show the results in Table 4. As can be seen, we outperform SensEmBERT in the four datasets from SemEval-2013, sometimes by a large margin, i.e., by almost 8 points on the Italian dataset. On SemEval-2015 we outperform SyntagRank by 6.1 points on the Spanish dataset and by 2.8 points on Italian one. We also show noticeable improvements over the baseline in 5 out of 6 benchmarks. The evaluation demonstrates that the EWISER approach is robust in the cross-lingual setting as well, outperforming competitors across the board and setting a new state of the art. Moreover, the results provide the empirical grounds for believing that, in addition to the results achieved in the languages featured in the evaluation datasets, comparable figures could also be attained for other languages, at least for several European ones.

5 Analysis

In this section we provide a qualitative analysis of our approach. Specifically, we are interested in the capability of the model to predict unseen synsets, thanks to the prior knowledge that is encoded in both the output embeddings O and the adjacency matrix A. Consider the following sentences:

(1) a. Corporate debt *defaults* predicted to increase.

b. Though people are free to change the *default*, they usually don't.

In Table 5 we report the predictions for the target *default* in sentences (1a) and (1b) of our best sys-

[4] github.com/SapienzaNLP/mwsd-datasets.

Synset	N	Gloss	w	**z** (1a)	**q** (1a)	**z** (1b)	**q** (1b)
`default.n.01`	1	loss due to not showing up	-	8.6	15.9	14.9	24.5
`loss.n.03`	6	the act of losing someone or something	.50	6.7	-	9.3	-
`absence.n.02`	8	failure to be present	.48	8.1	-	10.2	-
`default.n.02`	0	act of failing to meet a financial obligation	-	10.2	17.0	8.9	14.6
`default.v.01`	1	fail to pay up	.30	14.4	-	11.2	-
`failure.n.01`	18	an act that fails	.27	9.3	-	8.6	-
`nonpayment.n.02`	0	loss resulting from failure of a debt to be paid	-	11.0	**17.9**	9.6	15.5
`default.v.01`	1	fail to pay up	.30	14.4	-	11.2	-
`financial_loss.n.01`	0	loss of money or decrease in financial value	.29	8.7	-	8.6	-
`default_option.n.01`	0	an option that is selected automatically unless an alternative is specified	-	6.7	12.5	14.6	**25.5**
`option.n.02`	19	one of a number of things from which only one can be chosen	.76	7.7	-	14.3	-

Table 5: Predictions for sentences (1a) and (1b) of the best model trained on SemCor. In the first row of each block, we report the scores of the four synsets associated in WordNet with the noun *default*. The following rows contain the scores for synsets that are incident to those in the first row of the block, and contribute to their scores in **q**. The columns report, from left to right, a sense (therefore synset) identifier, the number of occurrences of that lemma in SemCor, the gloss, the weight of the edge, the hidden logits **z** and the output logits **q**.

tem trained on SemCor only, i.e., EWISER$_{hyper}$. In both cases, the correct synsets, respectively, `default.n.02`/`nonpayment.n.02` and `default_option.n.01`, are not in the training set. However, the model is still able to give the correct answer. In the first case, the embedding intialization is enough to predict `nonpayment.n.02` (with `default.n.02` having the second highest score), as its score in **z** is already the highest among possible predictions. In the latter, it is the contribution from the synset pointing to `default_option.n.01`, i.e., `option.n.02`, that enables the network to make the correct prediction.

However, we must note that the model still over-relies on corpus supervision. Because of this, even though our best overall model, i.e., EWISER$_{hyper}$ trained on SemCor, tagged glosses and examples, is able to distinguish and predict correctly the two well-attested mathematical meanings of *root* as equation solution and *root* as the number x such that $y = x^2$ in sentences (2a) and (2b) below, it is not able to correctly detect the tooth sense of root (2c), which never occurs in SemCor:

(2) a. The n *roots* of a polynomial of degree n depend continuously on the coefficients.

b. The *root* of 4 is 2.

c. There's no need to be worried if your dentist prescribes a *root* canal procedure.

Thus, while the EWISER model is indeed very effective, with the best configuration outdoing the upper bound on inter-annotator agreement, we are still far from having solved the task.

6 Conclusion

We presented EWISER, a new neural WSD architecture that, by embedding information from the WordNet graph within the neural architecture, can also make use of the relational information that is usually only exploited by knowledge-based systems. Thanks to the joint exploitation of the WordNet graph and to the use of pretrained synset embeddings, EWISER is able to predict meanings which are not found in the training set, thus mitigating the knowledge acquisition bottleneck.

On almost all the evaluation settings, our system beats the previous state of the art. Most notably, our model is the first to break through the 80 F1 ceiling on the overall evaluation, the estimated upper bound on the task. On the multilingual setting, even with no training data besides the English corpora, EWISER sets the new state of the art.

We leave it as future work to explore ways to raise accuracy on unseen synsets without harming performances on frequent synsets. We release the code used in the experiments, as well as pretrained models at `github.com/SapienzaNLP/ewiser`.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 under the European Union's Horizon 2020 research and innovation programme.

This work was supported in part by the MIUR under the grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of the Sapienza University of Rome.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece. Association for Computational Linguistics.

Michele Bevilacqua and Roberto Navigli. 2019. Quasi Bidirectional Encoder Representations from Transformers for word sense disambiguation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 122–131, Varna, Bulgaria. INCOMA Ltd.

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.

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

Steffen Eger, Paul Youssef, and Iryna Gurevych. 2018. Is it time to swish? Comparing deep learning activation functions across NLP tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4415–4424, Brussels, Belgium. Association for Computational Linguistics.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized 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)*, pages 5300–5309, Hong Kong, China. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. 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)*, pages 3500–3505, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.*

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife Hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. 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.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3525–3531, Hong Kong, China. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

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

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation*

(SemEval 2015), pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

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

Roberto Navigli. 2018. Natural Language Understanding: Instructions for (Present and Future) Use. In *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 5697–5702, Stockholm, Sweden.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

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

Tommaso Pasini. 2020. The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2020*. ijcai.org.

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.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. 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)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. Association for Computational Linguistics.

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

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. 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.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2018. Searching for activation functions. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track*.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. SensEmBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*.

Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Tolga Uslu, Alexander Mehler, Daniel Baumartz, and Wahed Hemati. 2018. FastSense: An efficient word sense disambiguation classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

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