Word Sense Disambiguation as a Game of Neurosymbolic Darts

Tiansi Dong, Rafet Sifa
Media Engineering, Fraunhofer IAIS
Schloss Birlinghoven, 1, 53757 Sankt Augustin, Germany
{tiansi.dong, rafet.sifa}@iais.frauhofer.de

Abstract

Word Sense Disambiguation (WSD) is one of the hardest tasks in natural language understanding and knowledge engineering. The glass ceiling of the 80% F1 score is recently achieved through supervised learning, enriched by knowledge graphs. Here, we propose a novel neurosymbolic methodology that may push the F1 score above 90%. The core of our methodology is a neurosymbolic sense embedding, in terms of a configuration of nested n-dimensional balls. The central point of a ball well preserves pre-trained word embeddings learned from data, which partially fixes the locations of balls. Inclusion relations among balls precisely encode symbolic hypernym relations among senses, and enable simple logic deduction among sense embeddings. We trained a Transformer to learn the mapping from a contextualized word embedding to its sense ball embedding, just like playing the game of darts (a game of shooting darts into a dartboard). A series of experiments are carried out using pretraining n ball embeddings, which cover around 70% training data and 75% testing data in the benchmark WSD corpus. Euclidean distance and cosine similarity functions are used as objective functions, separately, and each reaches $> 95.0\%$ F1 score in the ALL-$n$ball dataset. This substantially breaks the glass ceiling of deep learning methods. Future work is discussed to develop a full-fledged neurosymbolic WSD system that substantially outperforms deep learning approaches.

Keywords: sense disambiguation, neurosymbolic representation, knowledge graph, NLP

1. Introduction

Word Sense Disambiguation (WSD) is the task of acquiring the intended meaning of a word within the context where it appears (Navigli, 2009). It is one of the fundamental topics of natural language understanding in Artificial Intelligence (AI) (Weaver, 1949/1955), in part because WSD is hard, and has wide applications, such as information extraction, machine translation, opinion mining, question-answering, sentiment analysis, text understanding. Deep learning approaches have attained estimated human performance, and reached a glass ceiling over 80% (Bevilacqua et al., 2021), yet, they still make simple mistakes that humans would not do (Maru et al., 2022). Technically, classifying a word and its context into a word-sense class is limited to the knowledge that can be acquired from the training data (Bevilacqua et al., 2021), because word-senses are represented as *opaque* classes, and symbolic hypernym relations among senses cannot be used for deduction in the vector space. However, recent researches show ways to represent sense class in probabilistic box lattice (Vilnis et al., 2018) or fuzzy boxes (Dasgupta et al., 2022), or approximated in the hyperbolic space (Nickel and Kiela, 2017). However, it is possible to embed without loss a large symbolic tree-structured taxonomy of word senses as nested spheres with crisp boundaries, while well-preserving pre-trained vector embedding in the sphere centres (Dong et al., 2019a,b; Dong, 2021). In such a neurosymbolic paradigm, a word-sense is no more an *opaque* class; rather,

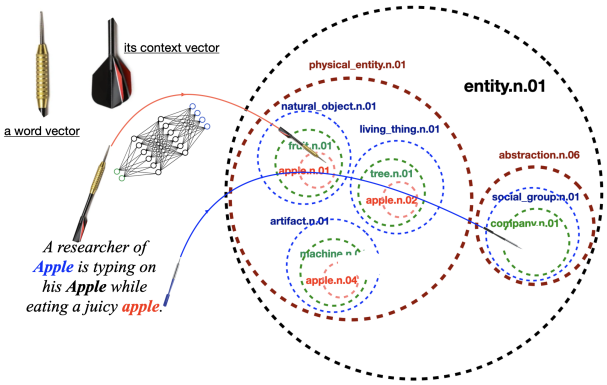

Figure 1: A neurosymbolic approach to Word-Sense Disambiguation works like playing the game of Dart. A deep neural network learns to shoot a contextualized word embedding vector to its sense regions in the Dart board.

it is explicitly embedded as an n-dimensional region with a crisp boundary. This provides a new way to tackle the tough WSD problem. Here, we vividly describe the new approach as a game of darts as follows: A neurosymbolic WSD is a neural dart player that shoots a contextualized word vector to the place of a configuration of regions, where its sense is located. This configuration of regions precisely encodes the sense inventory and latent features of words, as illustrated in Figure 1.
For example, apple.n.01, orange.n.01, and watermelon.n.01 are members of fruit.n.01. In classic deep-learning approaches, they are embedded as

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024, pages 22–32

four vectors. Here, we extend them into regions, so that their membership relations are explicitly represented by inclusion relations among these balls: the ball of fruit.n.01 contains the balls of apple.n.01, orange.n.01, and watermelon.n.01. The advantage for WSD is not only that shooting to a region is much easier than shooting to a point, but also that explicit region representation enables logical deduction among senses: shooting a contextualized vector of the word *apple* to the region of fruit.n.01 is sufficient to determine apple.n.01 as the intended sense; while shooting to the region of abstraction.n.06 is reasonable to hypothesize that it may refer to a company, even *apple* does not have the sense of company (*company.n.01*) in the sense inventory, as shown by the blue shooting path in Figure 1. The contribution in this paper are listed as follows.

1. We propose a novel neurosymbolic methodology for WSD, which seamlessly unifies supervised learning approaches and simple symbolic reasoning among hypernym relations;

2. We implement a simple Transformer to realise the first neurosymbolic WSD system, whose input is pre-trained word embeddings and whose output is a vectorial location in the pre-trained n-ball sense embeddings. The performance of this WSD breaks the ceiling of traditional deep-learning approaches in all 6 benchmark datasets where hypernym structures are available, and outperforms ChatGPT;

3. Our experiments show that using Direct Upper Hypernym (DUH) in testing achieved the best F1 score, while using DUH in training reduces the amount of training senses without weakening the performance;

4. Supported by our preliminary experiments, we envisage a novel neurosymbolic WSD system that may greatly outperform current SOTA systems and list a number of future works.

The rest of the article is structured as follows: we first review the recent WSD methods, and motivate our approach; then, we describe the details of the novel neurosymbolic approach. In experiments, we first set the targets, and report the statistics of training and testing dataset, then report and analyse experiments results, by comparing with performances of SOTA WSD systems and ChatGPT. In the end, we list a number of future work to realise full-fledged neurosymbolic WSD systems.

2. Related Works

2.1. Word Sense Disambiguation

The research on Word Sense Disambiguation (WSD) has a long history, with contributions from many researchers worldwide. A recent survey can be found in (Bevilacqua et al., 2021). The task of WSD is to automatically decide the intended sense in a given context, where senses of words are selected from the fixed word-sense inventory. A WSD system has three components, as follows: (1) a word in a given context, (2) a word sense inventory, e.g., WordNet (Miller et al., 1990; Miller, 1995), BabelNet (Navigli et al., 2021), and (3) an annotated corpus, e.g. SemCor (Miller et al., 1993), where some words have been manually or automatically annotated with intended word senses. The knowledge graph approaches and supervised deep-learning approaches are the main WSD approaches. Their performances are determined by the quality and the size of the knowledge bases (Pilehvar and Navigli, 2014).

Knowledge-based approaches for WSD Knowledge-based approaches leverage part of the graph structure of word-sense inventories, e.g. WordNet, BabelNet, where words connect with all their senses. By injecting the context of a word into the graph will slightly change the graph structure, and affect the probability distribution of senses of the word in the graph, which can be computed by the Personized PageRank algorithm (Agirre et al., 2014). The sense with the highest probability will be selected. This approach can be improved by connecting word-sense inventory with large web texts, e.g., BabelNet (Navigli et al., 2021), a knowledge base that integrates WordNet with Wikipedia (Moro et al., 2014).
From the game theoretical perspective (von Neumann and Morgenstern, 1947), a word can be viewed as a player, and its possible senses as strategies that the player can choose, to maximize a utility function (Tripodi and Navigli, 2019). Precisely, let $W = \{w_1, \ldots, w_n\}$ be the set of the content words in text T, $S_i = \{s_1, \ldots, s_{m_i}\}$ be the set of senses of w_i, $\mathcal{S} = \bigcup S_i$ is the set of all the strategies of the games. The strategy space of a player w_i is represented as a probabilistic distribution $\mathbf{x}_i$. The way how the context determines senses of words is simulated by interactions between two words w_i and w_j through a utility matrix Z. The cell $z_{r,t}$ represents the utility value when w_i chooses the r^{th} strategy and w_j chooses the t^{th} strategy. The value of one sense's strategy is related to its partners, in the following three aspects: word similarity, word-sense similarity, and their sense distributions, and computed in the manner similar to the attention mechanism.

Supervised deep learning for WSD Supervised deep learning approaches frame WSD as a multi-classification task – classifying a word w plus its context C into one of its word-senses s, using an annotated corpus $\mathcal{D}$, in the form of a list of triples $< w, c, s >$, and realized by supervised deep learn-

ing (Kågebäck and Salomonsson, 2016; Raganato et al., 2017b; Uslu et al., 2018).

The straightforward way of the supervised deep-learning approach is to compare the similarity between the contextualized embedding of a word w in the testing context c and senses s in the annotated corpus, and choose the most similar one, measured by a loss function $\mathcal{L}(w, c, s)$, either by feed-forward networks (Hadiwinoto et al., 2019), or transformers (Bevilacqua and Navigli, 2019). In these approaches, word senses are treated as discrete class labels. This may cause poor performance on low-frequency senses. To overcome this limitation, (Kumar et al., 2019) explicitly computed word sense embeddings by applying embedding methods for the hypernym structure of the WordNet, then trained an attentive BiLSTM to learn the context embedding of a word to its sense embedding. (Scarlini et al., 2020) computed contextualized sense embeddings by utilizing a variety of resources, such as SemCor, gloss in WordNet, SyntagNet (Maru et al., 2019), UKB (Agirre et al., 2014), and BERT (Devlin et al., 2018). (Loureiro and Jorge, 2019) computed sense embeddings by fully utilizing relations in WordNet, and achieved very competitive performance. Using explicit sense embeddings, (Bevilacqua and Navigli, 2020) successfully reached over 80% F1 score for WSD. (Barba et al., 2021) is able to choose the most important context definition for the target word. Their method inherits the idea of the game-theoretic WSD approach by using a feedback loop to consider the explicit senses of nearby words.

2.2. Neuosymbolic Unification

Both knowledge-based and supervised deep-learning WSD approaches have two assumptions as follows: (1) word senses are opaque classes, (2) a sense inventory has a fixed taxonomy (Bevilacqua et al., 2021). Consequently, in knowledge-based WSD approaches, word senses are represented by probabilistic distributions; in supervised WSD approaches, word senses are represented by latent vector embeddings. However, the two assumptions are somehow incompatible with the existence of a symbolic sense inventory – if a sense inventory has a well-structured and fixed taxonomy, why senses are opaque classes in both approaches? Such incompatibility lies in the discrepancy between the continuous numeric sense representation and the discrete symbolic sense representation – The continuous numeric representation, either as a probabilistic distribution or as a latent vector, cannot explicitly represent the well-defined symbolic taxonomy structure. This incompatibility could be resolved, if word sense embedding can precisely encode the discrete symbolic fixed taxonomy.

A vector sense embedding can be enlarged into an n-dimensional ball, whose radius is geometrically computed to strictly satisfy two conditions as follows: (1) balls of sibling senses are disconnected from each other; (2) balls of child and parent senses are precisely nested – the ball of a child sense is inside the ball of its parent sense. By utilising geometric methods, (Dong et al., 2019a) precisely injected a large tree-structured taxonomy of senses in WordNet-3.0 into pre-trained word embeddings, resulting in a configuration of nested low-dimensional balls. Thus, these nested balls unify numerical vector embeddings and symbolic structures into one representation without loss. Hyperbolic geometric embedding also has the power of neuro-symbolic unification (Tifrea et al., 2019; Chami et al., 2020), so that computational models can inherit good features from both neural computing and symbolic reasoning (Besold et al., 2017; Dong, 2021; Dong et al., 2022; Garcez and Lamb, 2023).

3. *Dart4WSD*: A neurosymbolic Darter

Dart4WSD is a novel supervised neurosymbolic learning methodology for Word Sense Disambiguation, with the novelty that senses are embedded as regions in vector space and that these region embeddings explicitly represent a fixed taxonomy in a sense inventory and well-preserve pre-train vector embeddings. *Dart4WSD* utilises a Transformer to learn the intended sense of a word in a given context, whose general architecture consists of five components; word embedding, a fixed sense inventory, a network that learns contextualized word embedding, a network that transforms the contextualized word embedding to a location in the neurosymbolic region, as illustrated in Figure 2.

3.1. Notations used in *Dart4WSD*

Let w and $\overrightarrow{w}$ be a word and its vector word-embedding, respectively, C represent a context; $\overrightarrow{w}_C$ be a vector embedding of word w in the context C. Let $\overrightarrow{V}_{w_C}$ be the output of our neural network, with the input $\overrightarrow{w}_C$, that is, $\overrightarrow{V}_{w_C} = NN(\overrightarrow{w}_C)$. Let w have k different senses in the inventory $\mathcal{S}_w = \{S_1^w, \ldots, S_k^w\}$, and $\mathcal{O}[S_i^w]$ be the ball embedding of S_i^w, with the central point $\overrightarrow{O}[S_i^w]$ and the radius $r[S_i^w]$.

3.2. The task formulation for *Dart4WSD*

Given an annotated corpus $\mathcal{D}$, we train a neural network NN, with a loss function $\mathcal{L}(NN(\overrightarrow{w}_C), \mathcal{O}[S_i^w])$ that improves the shooting technique of NN so that most of its output vectors are located inside balls of the target senses. In this preliminary work, we compare two objective functions: (1) the Euclidean distance, $\overrightarrow{V}_{w_C} = NN(\overrightarrow{w}_C)$ is inside $\mathcal{O}[S_i^w]$, that is, the distance between $\overrightarrow{V}_{w_C}$ and $\overrightarrow{O}[S_i^w]$ is less than or equal to $r[S_i^w]$. That is, $\mathcal{L}_{dis}(\overrightarrow{V}_{w_C}, \mathcal{O}[S_i^w]) = \max\{0, \|\overrightarrow{V}_{w_C} - \overrightarrow{O}[S_i^w]\| -$

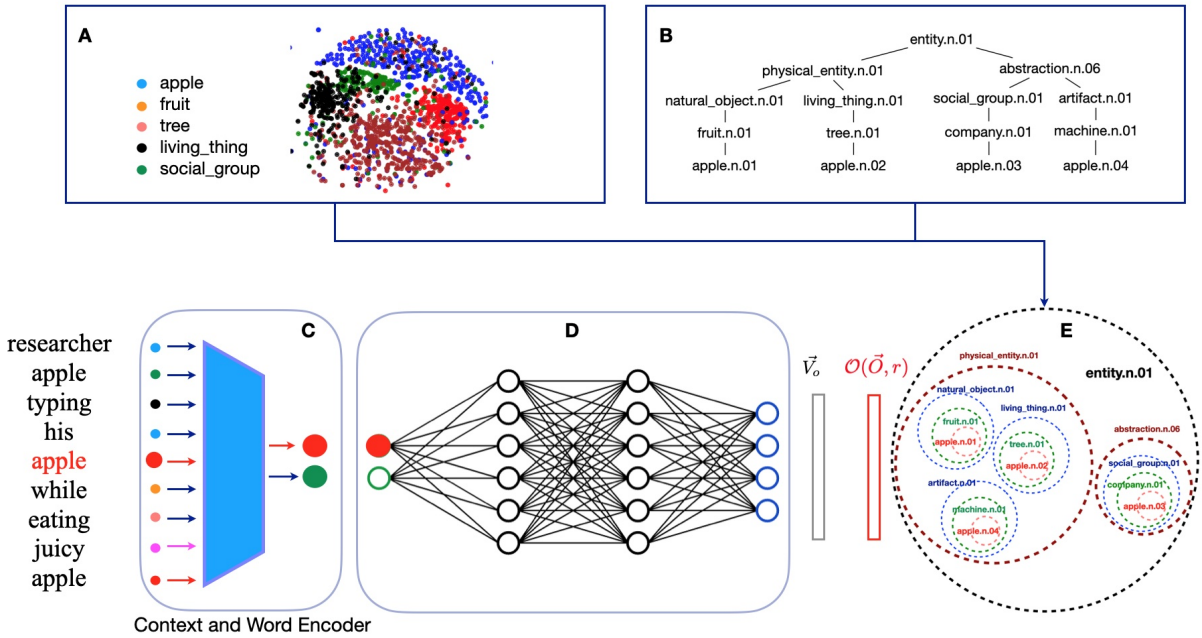

Figure 2: The supervised learning architecture of *Dart4WSD*: (**A**) word embeddings; (**B**) the fixed word sense taxonomy extracted from a sense inventory; (**C**) a neural network that learned contextualized word embeddings; (**D**) a neural network that learns to map a word in a context to its word sense ball embedding; (**E**) the neurosymbolic nested ball embeddings of word senses.

$r[S_i^w]\}$; (2) the well-known cosine similarity, that is, $\mathcal{L}_{cos}(\overrightarrow{V}_{w_C}, \mathcal{O}[S_i^w]) \approx \cos(\overrightarrow{w}_C, \overrightarrow{O}[S_i^w])$. The cosine approximation works well, when balls of sibling senses in the inventory are of the similar size. For example, in Figure 3, the apple.n.01, orange.n.01, and watermelon.n.01, three child senses of fruit.n.01, are embedded as balls with similar sizes; fruit.n.01 and tree.n.01 are siblings at the upper level in the inventory, and also embedded in the similar size. To correctly determine that the word *apple* in the phrase *eating a juicy apple*, the neural network shall map the contextualized word embedding ($\overrightarrow{apple}_{\text{eating a juicy}}$) to a vector inside the ball of the sense fruit.n.01 ($\mathcal{O}[S_1^{fruit}]$). Then, the sense apple.n.01 inside the fruit.n.01 will be chosen as the target sense.

Using upper category information for WSD in the embedding space has been proposed in (Beviá et al., 2006; Vial et al., 2019), we show that using explicit region embedding can fully utilise the upper category information, for at least two reasons as follows: (1) explicit and precise boundaries of regions endow our method the ability to reason with the symbolic hypernym relations in the embedding space; (2) It is reasonable to argue that the context information *eating a juicy…* shall not provide information to direct the word embedding of *apple* exactly to the ball embedding of apple.n.01, as *eating a juicy orange* and *eating a juicy watermelon* are as meaningful as *eating a juicy apple*. We argue that this context information shall direct the word

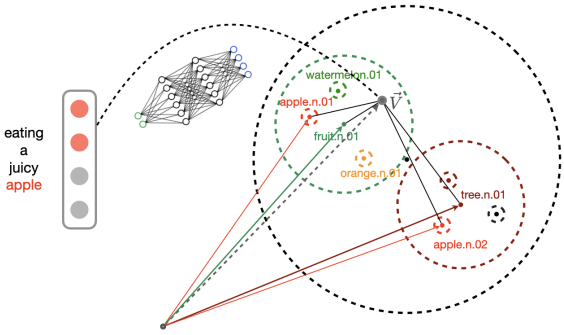

Figure 3: A novel method to choose senses by carrying out reasoning with hypernym relations in the embedding space: as long as the contextualised word embedding $\overrightarrow{apple}_{\text{eating a juicy}}$ is shot within the fruit.n.01 ball, our system will choose apple.n.01 as the target sense.

embedding of *apple* towards the sense embedding of its direct upper hypernym, here, fruit.n.01, and deviate from direct upper hypernym balls of its other senses, here, tree.n.01.

Let $H_1(S_i^w)$ be the direct upper hypernym of S_i^w in the inventory. We assume that there are no two S_i^w and S_j^w have the same direct upper hypernym, that is, $H_1(S_i^w) \neq H_1(S_j^w)$, if $S_i^w \neq S_j^w$. In the case of using Euclidean distance as the objective function, the sense of w, whose $\mathcal{O}[H_1(S_i^w)]$ (the boundary of the ball of the direct upper hypernym of w) is

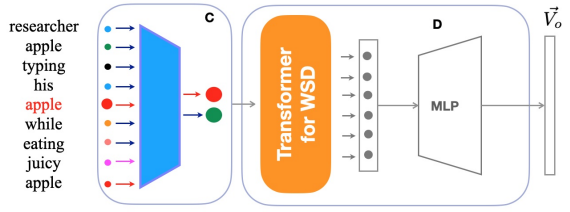

Figure 4: A transformer architecture for *Dart4WSD*.

nearest to $\overrightarrow{V}_{w_C}$, will be selected as the sense of w in the current context.

$$S^w = \arg \min_{S_i^w \in \mathcal{S}_w} \max\{0, \|\overrightarrow{V}_{w_C} - \vec{O}[S_i^w]\| - r[S_i^w]\}$$

In the case of using cosine similarity as the objective function, the sense of w, whose $\overrightarrow{O}[H_1(S_i^w)]$ (the centre vector of the ball of the direct upper hypernym of w) has the largest cosine value with $\overrightarrow{V}_{w_C}$, will be selected as the sense of w in the current context.

$$S^w = \arg \max_{S_i^w \in \mathcal{S}_w} \cos(\overrightarrow{V}_{w_C}, \overrightarrow{O}[H_1(S_i^w)])$$

3.3. The neurosymbolic Dartboard for senses

Considering the normal WSD situation that each contextualised word has one target sense, we restrict here the taxonomy of word senses as a tree structure. Accordingly, child-parent senses are precisely encoded as the child ball is inside the parent ball; sibling relation senses are precisely encoded as the disconnectedness relations among sibling sense balls, as illustrated in Figure 3. These features are fulfilled by n-ball embeddings (Dong et al., 2019a,b; Dong, 2021), in which (1) the symbolic taxonomy of word senses is explicitly and precisely encoded by boundary relations among regions, and (2) existing vector sense embedding is preserved by the centre vector of a region, as illustrated in Figure 2. Thus, we use n-balls as the neurosymbolic Dartboard of *Dart4WSD*, for quick prototyping and the proof of concept, and also for the ease of re-production and extension.

3.4. A supervised learning process

The Transformer architecture was originally designed for sequence-to-sequence tasks (Vaswani et al., 2017), and has been applied in a variety of fields (Lin et al., 2021). It can be used as a universal approximation of sequence-to-sequence functions (Yun et al., 2020). We use a Transformer architecture to learn the mapping from the contextualized words to balls of their target senses, as illustrated in Figure 4. Given a sentence s, we transform it into a list of tokens $(t_1, t_2, t_3 ... t_m)$, then, replace each token with contextualised word embedding,

	#training	#exclude	#n-ball	#no ball
SC	224415	56207	156483	11725
SC+O	1135547	259375	837147	39025

Table 1: The statistics of the numbers of training records. **SC** represents **SemCor**; **SC**+**O** represents **SemCor**+**OMSTI**.

senses	#s-class	#s-nball	#s-L1
SC	18953	**15025**	**5799**
SC+O	19253	**15298**	**5852**

Table 2: The statistics of senses in our experiments. #s-class: the total number of senses whose hypernym path is longer than 1; #s-nball: the total number of senses that have ball embedding; #s-L1: the total number of senses that are the direct hypernym of senses in #s-nball.

$\overrightarrow{t}_{1,C_1}, \overrightarrow{t}_{2,C_2}, \overrightarrow{t}_{3,C_3} \ldots \overrightarrow{t}_{m,C_m}$ (Yap et al., 2020). We feed $\overrightarrow{t}_{C_i}$ into a Transformer (TF), whose outputs are fed into a two-layered perceptron as follows. Ideally, the output of the perceptron $\overrightarrow{V}$ shall be inside the n-ball of the target word sense.

$$\overrightarrow{V}_{w_C} = Linear(Relu(Linear(TF(\overrightarrow{t}_{i,C_i}))))$$

4. Experiments

The target of the experiments is to examine the WSD performance, when the symbolic structure of the sense classes is explicitly and precisely represented in the vector space. We developed *Dart4WSD* as the first such a WSD system, and compared its performance with the SOTA performance, and with the WSD performance of ChatGPT. Our four experiments are designed to answer the questions as follows.

1. How is the WSD performance of LLMs, e.g., ChatGPT?

2. How good is *Dart4WSD* in the task of mapping contextualized word **vector** to sense **vectors**, using Euclidean distance and cosine similarity objective functions, respectively? Which objective function leads to better performance?

3. How is the performance of *Dart4WSD*, if it uses the direct upper hypernym of the target sense (here, n-dimensional balls)? Which objective function leads to better performance?

4. Will the performance be improved in the testing phase, if in the learning phase *Dart4WSD* maps to n-balls of direct upper hypernym senses?

	#test	#exclude	#nball	#no nball
S-2	2275	722	1459	94
S-3	1832	396	1341	95
S-07	449	8	420	21
S-13	1621	0	1435	186
S-15	1013	248	712	53
ALL	7181	1374	5358	449

Table 3: The statistics of testing records. **S-2** represents **Senseval-2**, **S-3** represents **Senseval-3**, **S-07** represents **SemEval-07**, **S-13** represents **SemEval-13**, **S-15** represents **SemEval-15**.

	#S-2/L1	**#S-3/L1**	**#S-07/L1**
#nball	711/522	780//605	327/281
	#S-13/L1	**#S-15/L1**	**#A/L1**
#nball	669/408	350/256	2251/1424

Table 4: The statistics of senses in test records. **#S-2/L1** represents the numbers of different n-balls in **Senseval-2** and the direct upper level hypernyms. Others are interpreted in the same way.

4.1. Datasets

We exclude, from benchmark datasets SemCor (**SC**) and SemCor+OMSTI (**SC+O**), senses that do not have class structures, as our target focuses on the WSD performances, subject to *opaque* or *clear* embedding of sense classes.

SemCor has 224415 training records, among which there are 25845 different senses; senses in 156483 records have n-ball embedding, among which there are 15025 different senses; senses in 11725 records do not yet have n-ball embedding, totalling 3928 different senses. Senses in 56207 records do not have a taxonomy, totalling 6892 different senses.

SemCor+OMSTI has 1135547 training records, among which there are 26265 different senses; senses in 837147 records have n-ball embedding, among which there are 15298 different senses. Senses in 39025 records do not yet have n-ball embedding, totaling 3955 different senses. Senses in 259375 records do not have a taxonomy, totaling 7012 different senses, as listed in Table 1 and Table 2. The n-ball embedding contains 47,634 word senses, covering around 80% senses in the WSD benchmark datasets.

4.1.1. Training data

We create four training datasets, as follows: (1) **SemCor-$n$ball**, (2) **SemCor+OMSTI-$n$ball**, (3) **SemCor-$n$ball-L1**, and (4) **SemCor+OMSTI-$n$ball-L1** in the following way: Firstly, we transform training data into the form as follows: "(sense, a list of word, the index for the word(s) of the sense)". For example, *('aim.n.02', ['have', 'you', 'set', 'specific', 'objectives'], [4])*, which means that the word pointed by the index 4, that is the word 'objectives', should have the sense 'aim.n.02'. The first two datasets **SemCor-$n$ball** and **SemCor+OMSTI-$n$ball** are extracted from SemCor and SemCor+OMSTI with the criteria that target senses have n-ball embeddings. That is, if 'aim.n.02' has an n-ball embedding, this piece of training record will be selected. The other two datasets are created, by setting each target sense in the first two datasets with its direct hypernym. If this hypernym has n-ball embedding, the training record will be selected. For example, 'aim.n.02' has an hypernym path in WordNet-3.0, as follows: ['aim.n.02', 'goal.n.01', 'content.n.05', 'cognition.n.01', ...]. Its direct hypernym is 'goal.n.01'. If it has an n-ball embedding, the following training record will be added into the corresponding **-L1** dataset, for example, *('goal.n.01', 'aim.n.02', ['have', 'you', ..., 'objectives'], [4])*.

4.1.2. Testing data

We create $6 \times 2 = 12$ datasets from the six benchmark datasets, namely, **Senseval-2**, **Senseval-3**, **SemEval-07**, **SemEval-13**, **SemEval-15**, **ALL** (Raganato et al., 2017a). From each dataset **E**$\in$**{Senseval-2**, **Senseval-3**, **SemEval-07**, **SemEval-13**, **SemEval-15**, **ALL}**, we derive 2 testing datasets as follws: **E-$n$ball** and **E-$n$ball-L1**. **E-$n$ball** and **E-$n$ball-L1** are created in the same way as we create training data, as listed in Table 3.

4.1.3. Evaluation

We use the F1 calculation software in the standard WSD corpus, downloaded from `http://lcl.uniroma1.it/wsdeval/home`.

4.2. Setting and running of experiments

Dart4WSD is implemented in PyTorch. We set learning rate to 0.001, 20 epochs, with 4-fold cross validation. Experiments were conducted on MacBook Pro Apple M1 Max (10C CPU/24C GPU), 32 GB memory. Using 50-d Glove word embedding, *Dart4WSD* took less than 10 seconds for one epoch for SemCor-$n$ball training data. *Dart4WSD* converges very fast: the loss of the second epoch is only one tenth of the loss of the first epoch.

4.3. Experiments and Results

4.3.1. Experiment 1

Recent research shows that LLMs, e.g., ChatGPT, can do almost perfect human-like question-answering, and their ability to reason can be improved by using prompt engineering. We created four kinds of prompts to evaluate performances of ChatGPT (gpt-3.5-turbo) on the six benchmark WSD test datasets, as follows: (1) Zero-shot prompt, which gives ChatGPT all senses of a word w, and a sentence containing w, and let ChatGPT choose the right one from the list; (2) few-shot

Obj. func.:	Senseval-2		Senseval-3		Senseval-07		Senseval-13		Senseval-15	
$\mathcal{L}_{dis}$	L0	L1	L0	L1	L0	L1	L0	L1	L0	L1
SC	34.1%	94.9%	37.5%	94.1%	32.0%	88.2%	32.8%	100.0%	33.9%	91.7%
SC L1	34.1%	**94.9%**	37.5%	**94.1%**	32.0%	**88.2%**	32.8%	**100.0%**	33.9%	**91.7%**
SC+O	34.1%	94.9%	37.5%	94.1%	32.0%	88.2%	32.8%	100.0%	33.9%	91.7%
SC+O L1	34.1%	94.9%	37.5%	94.1%	32.0%	88.2%	32.8%	100.0%	33.9%	91.7%

Table 5: F1 scores of 5×2 datasets by using Euclidean distance as the objective function. The F1 is computed by the standard tool for WSD, which is available in the dataset download from http://lcl.uniroma1.it/wsdeval/home.

Obj. func.:	Senseval-2		Senseval-3		Senseval-07		Senseval-13		Senseval-15	
$\mathcal{L}_{cos}$	L0	L1	L0	L1	L0	L1	L0	L1	L0	L1
SC	34.6%	95.2%	38.0%	93.3%	33.2%	89.5%	41.3%	100.0%	39.5%	92.9%
SC L1	34.6%	**95.2%**	38.0%	**93.3%**	33.2%	**89.5%**	41.3%	**100.0%**	39.5%	**92.9%**
SC+O	34.6%	95.2%	38.0%	93.3%	33.2%	89.5%	41.3%	100.0%	39.5%	92.9%
SC+O L1	34.6%	95.2%	38.0%	93.3%	33.2%	89.5%	41.3%	100.0%	39.5%	92.9%

Table 6: F1 scores of 5×2 datasets by using cosine similarity as the objective function.

prompt, which adds one example to the zero-shot prompt; (3) CoT prompt, which uses the gloss as a mid-step to connect a sense and the word in a context; (4) few-shot CoT, which adds an example to the Cot prompt. The zero-shot prompt produces the lowest performance, ranging from 30.7% to 37.6%, the few-shot CoT delivers the best performance, ranging from 55.4% to 68.4%, which is below 80% the glass ceiling of the SOTA performance. Other experiments found that LLMs may make correct answers with incorrect explanations (Creswell et al., 2022; Zelikman et al., 2022). Similarly, the case of WSD may provide chances to explore how ChatGPT may correctly understand the meaning of sentences, while misunderstanding the meanings of single words in the sentence.

4.3.2. Experiment2

To answer the second question, we used the **SemCor-$n$ball** dataset to train our *Dart4WSD* neural-network. It learns to map from contextualized word embeddings to centre vectors of sense n-balls. The performances using Euclidean distance range from 32.8% to 37.5% (F1 score); while the performances using cosine similarity range from 33.2% to 39.5%, in all the testing datasets, as illustrated in column L0 of Table 5, Table 6. Compared with the current best result 80% (Bevilacqua and Navigli, 2020), this performance is not good, in part because our inputs are pre-trained glove vectors and the context vector is approximated by averaging the vectors of neighbourhood words with a fixed window size, which limits the Transformer to dynamically select the right contexts, and results in a similar performance as ChatGPT using zero-shot prompt.

	ALL ($\mathcal{L}_{dis}$)		ALL ($\mathcal{L}_{cos}$)	
	L0	L1	L0	L1
SC	34.4%	95.2%	37.8%	95.3%
SC L1	34.4%	**95.2%**	37.8%	**95.3%**
SC+O	34.4%	95.2%	37.8%	95.3%
SC+O L1	34.4%	95.2%	37.8%	95.3%

Table 7: F1 scores of the ALL-L0 and ALL-L1 datasets. Using direct hypernyms of target senses (ALL-L1), the performances (with both objective functions) of Dart4WSD break the glass ceiling of deep learning methods.

4.3.3. Experiment3

For the third question, we used the trained model in Experiment 2, and evaluated whether it successfully hit the ball of the direct upper hypernym senses. The F1 scores range from 88.2% to 100% using Euclidean distance, and range from 89.5% to 100% using cosine similarity, as listed in Table 5, Table 6. The F1 score for the ALL-L1 data set reaches 95.0% (Table 7) with each objective function (Euclidean distance and cosine similarity), which greatly outperforms the SOTA performance (80%) (Bevilacqua et al., 2021), and break the performance ceiling (a bit above 90%) of traditional deep-learning approaches (Raganato et al., 2017a).

4.3.4. Experiment4

To answer the third question, we trained *Dart4WSD* by utilising the **SemCor-$n$ball-L1** and **SemCor+OMSTI-$n$ball L1** datasets. The target senses in the two training data sets are replaced by their direct hypernyms, so they have less number of senses for learning. There are no drops in the performance, as illustrated in the rows **SC L1** and **SC+O L1** of Table 5 – 7. This shows that *Dart4WSD* is less data-hungry, compared with

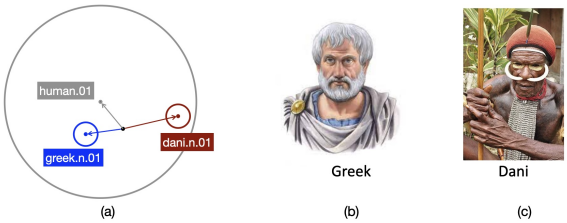

Figure 5: (a) the sphere boundary of human.n.01 includes the dani.n.01 sphere; (b-c) Sample images of Greek people and Dani people.

traditional deep learning systems.

4.3.5. Analysis and Discussions

By using the direct upper-level senses in the testing datasets, Dart4WSD outperforms ChatGPT and the SOTA systems, and even breaks the glass ceiling of deep learning approaches, in the setting of current experiments. We also performed experiments by utilising other pre-trained embeddings, e.g., BERT, and had very similar results. This convergently suggests that the high performance shall be ascribed to the neurosymbolic n-ball embedding that precisely imposes a symbolic sense inventory into the embedding space, while preserving pre-trained word embeddings in the centre points of these n-balls. In this way, the configuration of all n-balls endows *Dart4WSD* with the capability to better represent out-of-distribution data by utilising boundary relations among n-balls. For example, Dani people may have different cultures and histories from many other human races, e.g., Greeks. Their sample images as illustrated in Figure 5(b-c). Descriptions about them may appear in different types of corpus, which may result in different vector embeddings whose cosine similarity is less than 0, as illustrated in Figure 5(a). By utilising n-ball representation, they are represented within the human ball. This may bring the advantage to *Dart4WSD*, easier to make correct decisions, compared with traditional deep learning systems.

5. Conclusions and Outlooks

We prototyped *Dart4WSD*, a novel supervised neurosymbolic method for Word-Sense Disambiguation that dramatically outperforms the traditional deep learning approaches. The core of our method is a configuration of n-dimensional sphere embeddings whose boundary relations explicitly and precisely embed a symbolic sense inventory in the vector space and whose centre hosts latent features learned from data. This neurosymbolic approach is independent of languages and could be especially useful for low-resource languages. To this end, a number of problems shall be solved, listed as follows.

New Datasets for Neuro-symbolic WSD A benchmark dataset for neuro-symbolic WSD shall consist of not only labelled data for traditional supervised learning, but also a symbolic taxonomy of sense inventory. This symbolic part can be a part of a large sense inventory that only describes the taxonomy of senses in the labelled data.

Using a traditional deep-learning system as the backbone Our neurosymbolic method demonstrates its performance only when a well-designed sense inventory is available, which can be unrealistic. It would be promising to build up a neurosymbolic component above a traditional deep-learning WSD system.

More powerful geometric objective functions We used Euclidean distances and cosine similarity as two objective functions. Intuitively, Euclidean distance is more precise to measure relations between spheres, however, its performance in current experiments is a bit less than that of using cosine similarity, which cannot take the boundary information of balls into consideration. There should be powerful geometric objective functions to outperform the cosine similarity measurement.

N-ball for DAG structures The sense inventory in Word-Net 3.0 (Miller, 1995) is not a tree structure, but a Directed Acyclic Graph (DAG). We shall extend the current geometric approach for DAG structures. Creating a new n-ball configuration is not trivial, as the sense taxonomy needs to be precisely embedded (reaching the global loss of zero). This is a very challenging machine-learning task that is worth further research.

Heterogenous structure One assumption of our approach is that senses of word shall have different direct upper hypernyms, so, we can use balls of direct upper hypernyms. This assumption holds for nouns in most of the cases, but, might not hold for verbs. For example, fly.v.01 (*travel through the air; be airborne*) and fly.v.06 (*be dispersed or disseminated*) are both senses the word fly, they share the same direct upper hypernym travel.n.01 (*change location; move, travel, or proceed, also metaphorically*). In this case, using direct upper hypernym is not sufficient to disambiguate between fly.v.01 and fly.v.06. We may need to integrate other knowledge into the sense inventory. We may need to consider Descartes's product of n-balls. For example, one encodes hypernym relations, another encodes part-whole relations.

Towards a new methodology for classification *Dart4WSD* can be generalised for solving any classification problem. In contrast to traditional supervised deep-learning methods, our method will create the dart board before shooting, instead of the other way around (shooting first, then drawing the best-fit target, as described in (Gigerenzer, 2022)).

6. Bibliographical References

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

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021. Consec: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503.

Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR*, abs/1711.03902.

Michele Bevilacqua and Roberto Navigli. 2019. Quasi bidirectional encoder representations from transformers for word sense disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 122–131, Varna, Bulgaria. INCOMA Ltd.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. pages 4330–4338.

Rubén Beviá, Armando Suárez Cueto, and German Rigau. 2006. Exploring the automatic selection of basic level concepts.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning.

Shib Dasgupta, Michael Boratko, Siddhartha Mishra, Shriya Atmakuri, Dhruvesh Patel, Xiang Li, and Andrew McCallum. 2022. Word2Box: Capturing set-theoretic semantics of words using box embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2263–2276, Dublin, Ireland. 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.

Tiansi Dong. 2021. *A Geometric Approach to the Unification of Symbolic Structures and Neural Networks*, volume 910 of *Studies in Computational Intelligence*. Springer-Nature.

Tiansi Dong, Chrisitan Bauckhage, Hailong Jin, Juanzi Li, Olaf H. Cremers, Daniel Speicher, Armin B. Cremers, and Jörg Zimmermann. 2019a. Imposing Category Trees Onto Word-Embeddings Using A Geometric Construction. In *ICLR-19*, New Orleans, USA. May 6-9.

Tiansi Dong, Achim Rettinger, Jie Tang, Barbara Tversky, and Frank van Harmelen. 2022. Structure and Learning (Dagstuhl Seminar 21362). *Dagstuhl Reports*, 11(8):11–34.

Tiansi Dong, Zhigang Wang, Juanzi Li, Christian Bauckhage, and Armin B. Cremers. 2019b. Triple Classification Using Regions and Fine-Grained Entity Typing. In *AAAI-19*, pages 77–85.

Artur Garcez and Luís Lamb. 2023. Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review*, pages 1–20.

Gerd Gigerenzer. 2022. *How to Stay Smart in a Smart World: Why Human Intelligence Still Beats Algorithms*. The MIT Press.

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 5297–5306, Hong Kong, China. Association for Computational Linguistics.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional LSTM. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 51–56, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2021. A survey of transformers. *arXiv preprint arXiv:2106.04554*.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. pages 5682–5691.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. Syntagnet: Challenging supervised word sense disambiguation with lexical-semantic combinations. pages 3525–3531.

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

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database*. *International Journal of Lexicography*, 3(4):235–244.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA. 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 Computing Surveys*, 41(2).

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of babelnet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Nips*, pages 6338–6347.

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

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.

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.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. Association for Computational Linguistics.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincaré glove: Hyperbolic word embeddings. *ICLR-19*.

Rocco Tripodi and Roberto Navigli. 2019. Game theory meets embeddings: a unified framework for word sense disambiguation. pages 88–99.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through

the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference*, pages 108–117, Wroclaw, Poland. Global Wordnet Association.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272, Melbourne, Australia. Association for Computational Linguistics.

J. von Neumann and O. Morgenstern. 1947. *Theory of games and economic behavior*. Princeton University Press.

Warren Weaver. 1949/1955. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA. Reprinted from a memorandum written by Weaver in 1949.

Boon Yap, Andrew Koh, and Eng Chng. 2020. Adapting bert for word sense disambiguation with gloss selection objective and example sentences. pages 41–46.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Are transformers universal approximators of sequence-to-sequence functions? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

7. Language Resource References