Meta-Learning with Variational Semantic Memory for Word Sense Disambiguation

Yingjun Du
University of Amsterdam
y.du@uva.nl

Nithin Holla
Amberscript
nithin.holla7@gmail.com

Xiantong Zhen
University of Amsterdam
x.zhen@uva.nl

Cees G.M. Snoek
University of Amsterdam
C.G.M.Snoek@uva.nl

Ekaterina Shutova
University of Amsterdam
e.shutova@uva.nl

Abstract

A critical challenge faced by supervised word sense disambiguation (WSD) is the lack of large annotated datasets with sufficient coverage of words in their diversity of senses. This inspired recent research on few-shot WSD using meta-learning. While such work has successfully applied meta-learning to learn new word senses from very few examples, its performance still lags behind its fully-supervised counterpart. Aiming to further close this gap, we propose a model of semantic memory for WSD in a meta-learning setting. Semantic memory encapsulates prior experiences seen throughout the lifetime of the model, which aids better generalization in limited data settings. Our model is based on hierarchical variational inference and incorporates an adaptive memory update rule via a hypernetwork. We show our model advances the state of the art in few-shot WSD, supports effective learning in extremely data scarce (e.g. one-shot) scenarios and produces meaning prototypes that capture similar senses of distinct words.

1 Introduction

Disambiguating word meaning in context is at the heart of any natural language understanding task or application, whether it is performed explicitly or implicitly. Traditionally, word sense disambiguation (WSD) has been defined as the task of explicitly labeling word usages in context with sense labels from a pre-defined sense inventory. The majority of approaches to WSD rely on (semi-)supervised learning (Yuan et al., 2016; Raganato et al., 2017a,b; Hadiwinoto et al., 2019; Huang et al., 2019; Scarlini et al., 2020; Bevilacqua and Navigli, 2020) and make use of training corpora manually annotated for word senses. Typically, these methods require a fairly large number of annotated training examples per word. This problem is exacerbated by the dramatic imbalances in sense frequencies, which further increase the need for annotation to capture a diversity of senses and to obtain sufficient training data for rare senses.

This motivated recent research on few-shot WSD, where the objective of the model is to learn new, previously unseen word senses from only a small number of examples. Holla et al. (2020a) presented a meta-learning approach to few-shot WSD, as well as a benchmark for this task. Meta-learning makes use of an episodic training regime, where a model is trained on a collection of diverse few-shot tasks and is explicitly optimized to perform well when learning from a small number of examples per task (Snell et al., 2017; Finn et al., 2017; Triantafillou et al., 2020). Holla et al. (2020a) have shown that meta-learning can be successfully applied to learn new word senses from as little as one example per sense. Yet, the overall model performance in settings where data is highly limited (e.g. one- or two-shot learning) still lags behind that of fully supervised models.

In the meantime, machine learning research demonstrated the advantages of a memory component for meta-learning in limited data settings (Santoro et al., 2016a; Munkhdalai and Yu, 2017a; Munkhdalai et al., 2018; Zhen et al., 2020). The memory stores general knowledge acquired in learning related tasks, which facilitates the acquisition of new concepts and recognition of previously unseen classes with limited labeled data (Zhen et al., 2020). Inspired by these advances, we introduce the first model of semantic memory for WSD in a meta-learning setting. In meta-learning, prototypes are embeddings around which other data points of the same class are clustered (Snell et al., 2017). Our semantic memory stores prototypical representations of word senses seen during training, generalizing over the contexts in which they are used. This rich contextual information aids in learning new senses of previously unseen words

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5254–5268
August 1–6, 2021.

that appear in similar contexts, from very few examples.

The design of our prototypical representation of word sense takes inspiration from prototype theory (Rosch, 1975), an established account of category representation in psychology. It stipulates that semantic categories are formed around prototypical members, new members are added based on resemblance to the prototypes and category membership is a matter of degree. In line with this account, our models learn prototypical representations of word senses from their linguistic context. To do this, we employ a neural architecture for learning probabilistic class prototypes: variational prototype networks, augmented with a variational semantic memory (VSM) component (Zhen et al., 2020).

Unlike deterministic prototypes in prototypical networks (Snell et al., 2017), we model class prototypes as distributions and perform variational inference of these prototypes in a hierarchical Bayesian framework. Unlike deterministic memory access in memory-based meta-learning (Santoro et al., 2016b; Munkhdalai and Yu, 2017a), we access memory by Monte Carlo sampling from a variational distribution. Specifically, we first perform variational inference to obtain a latent memory variable and then perform another step of variational inference to obtain the prototype distribution. Furthermore, we enhance the memory update of vanilla VSM with a novel adaptive update rule involving a hypernetwork (Ha et al., 2016) that controls the weight of the updates. We call our approach β-VSM to denote the adaptive weight β for memory updates.

We experimentally demonstrate the effectiveness of this approach for few-shot WSD, advancing the state of the art in this task. Furthermore, we observe the highest performance gains on word senses with the least training examples, emphasizing the benefits of semantic memory for truly few-shot learning scenarios. Our analysis of the meaning prototypes acquired in the memory suggests that they are able to capture related senses of distinct words, demonstrating the generalization capabilities of our memory component. We make our code publicly available to facilitate further research.[1]

2 Related work

Word sense disambiguation Knowledge-based approaches to WSD (Lesk, 1986; Agirre et al., 2014; Moro et al., 2014) rely on lexical resources such as WordNet (Miller et al., 1990) and do not require a corpus manually annotated with word senses. Alternatively, supervised learning methods treat WSD as a word-level classification task for ambiguous words and rely on sense-annotated corpora for training. Early supervised learning approaches trained classifiers with hand-crafted features (Navigli, 2009; Zhong and Ng, 2010) and word embeddings (Rothe and Schütze, 2015; Iacobacci et al., 2016) as input. Raganato et al. (2017a) proposed a benchmark for WSD based on the SemCor corpus (Miller et al., 1994) and found that supervised methods outperform the knowledge-based ones.

Neural models for supervised WSD include LSTM-based (Hochreiter and Schmidhuber, 1997) classifiers (Kågebäck and Salomonsson, 2016; Melamud et al., 2016; Raganato et al., 2017b), nearest neighbour classifier with ELMo embeddings (Peters et al., 2018), as well as a classifier based on pretrained BERT representations (Hadiwinoto et al., 2019). Recently, hybrid approaches incorporating information from lexical resources into neural architectures have gained traction. GlossBERT (Huang et al., 2019) fine-tunes BERT with WordNet sense definitions as additional input. EWISE (Kumar et al., 2019) learns continuous sense embeddings as targets, aided by dictionary definitions and lexical knowledge bases. Scarlini et al. (2020) present a semi-supervised approach for obtaining sense embeddings with the aid of a lexical knowledge base, enabling WSD with a nearest neighbor algorithm. By further exploiting the graph structure of WordNet and integrating it with BERT, EWISER (Bevilacqua and Navigli, 2020) achieves the current state-of-the-art performance on the benchmark by Raganato et al. (2017a) – an F1 score of 80.1%.

Unlike few-shot WSD, these works do not fine-tune the models on new words during testing. Instead, they train on a training set and evaluate on a test set where words and senses might have been seen during training.

Meta-learning Meta-learning, or learning to learn (Schmidhuber, 1987; Bengio et al., 1991; Thrun and Pratt, 1998), is a learning paradigm where a model is trained on a distribution of tasks so as to enable rapid learning on new tasks. By solving a large number of different tasks, it aims to leverage the acquired knowledge to learn new, unseen tasks. The training set, referred to as the

[1] https://github.com/YDU-uva/VSM_WSD

meta-training set, consists of *episodes*, each corresponding to a distinct task. Every episode is further divided into a *support set* containing just a handful of examples for learning the task, and a *query set* containing examples for task evaluation. In the meta-training phase, for each episode, the model adapts to the task using the support set, and its performance on the task is evaluated on the corresponding query set. The initial parameters of the model are then adjusted based on the loss on the query set. By repeating the process on several episodes/tasks, the model produces representations that enable rapid adaptation to a new task. The test set, referred to as the *meta-test set*, also consists of episodes with a support and query set. The meta-test set corresponds to new tasks that were not seen during meta-training. During meta-testing, the meta-trained model is first fine-tuned on a small number of examples in the support set of each meta-test episode and then evaluated on the accompanying query set. The average performance on all such query sets measures the few-shot learning ability of the model.

Metric-based meta-learning methods (Koch et al., 2015; Vinyals et al., 2016; Sung et al., 2018; Snell et al., 2017) learn a kernel function and make predictions on the query set based on the similarity with the support set examples. Model-based methods (Santoro et al., 2016b; Munkhdalai and Yu, 2017a) employ external memory and make predictions based on examples retrieved from the memory. Optimization-based methods (Ravi and Larochelle, 2017; Finn et al., 2017; Nichol et al., 2018; Antoniou et al., 2019) directly optimize for generalizability over tasks in their training objective.

Meta-learning has been applied to a range of tasks in NLP, including machine translation (Gu et al., 2018), relation classification (Obamuyide and Vlachos, 2019), text classification (Yu et al., 2018; Geng et al., 2019), hypernymy detection (Yu et al., 2020), and dialog generation (Qian and Yu, 2019). It has also been used to learn across distinct NLP tasks (Dou et al., 2019; Bansal et al., 2019) as well as across different languages (Nooralahzadeh et al., 2020; Li et al., 2020). Bansal et al. (2020) show that meta-learning during self-supervised pre-training of language models leads to improved few-shot generalization on downstream tasks.

Holla et al. (2020a) propose a framework for few-shot word sense disambiguation, where the goal is to disambiguate new words during meta-testing. Meta-training consists of episodes formed from multiple words whereas meta-testing has one episode corresponding to each of the test words. They show that prototype-based methods – prototypical networks (Snell et al., 2017) and first-order ProtoMAML (Triantafillou et al., 2020) – obtain promising results, in contrast with model-agnostic meta-learning (MAML) (Finn et al., 2017).

Memory-based models Memory mechanisms (Weston et al., 2014; Graves et al., 2014; Krotov and Hopfield, 2016) have recently drawn increasing attention. In memory-augmented neural network (Santoro et al., 2016b), given an input, the memory read and write operations are performed by a controller, using soft attention for reads and least recently used access module for writes. Meta Network (Munkhdalai and Yu, 2017b) uses two memory modules: a key-value memory in combination with slow and fast weights for one-shot learning. An external memory was introduced to enhance recurrent neural network in Munkhdalai et al. (2019), in which memory is conceptualized as an adaptable function and implemented as a deep neural network. Semantic memory has recently been introduced by Zhen et al. (2020) for few-shot learning to enhance prototypical representations of objects, where memory recall is cast as a variational inference problem.

In NLP, Tang et al. (2016) use content and location-based neural attention over external memory for aspect-level sentiment classification. Das et al. (2017) use key-value memory for question answering on knowledge bases. Mem2Seq (Madotto et al., 2018) is an architecture for task-oriented dialog that combines attention-based memory with pointer networks (Vinyals et al., 2015). Geng et al. (2020) propose Dynamic Memory Induction Networks for few-shot text classification, which utilizes dynamic routing (Sabour et al., 2017) over a static memory module. Episodic memory has been used in lifelong learning on language tasks, as a means to perform experience replay (d'Autume et al., 2019; Han et al., 2020; Holla et al., 2020b).

3 Task and dataset

We treat WSD as a word-level classification problem where ambiguous words are to be classified into their senses given the context. In traditional WSD, the goal is to generalize to new contexts of word-sense pairs. Specifically, the test set consists of word-sense pairs that were seen during train-

ing. On the other hand, in few-shot WSD, the goal is to generalize to new words and senses altogether. The meta-testing phase involves further adapting the models (on the small support set) to new words that were not seen during training and evaluates them on new contexts (using the query set). It deviates from the standard N-way, K-shot classification setting in few-shot learning since the words may have a different number of senses and each sense may have different number of examples (Holla et al., 2020a), making it a more realistic few-shot learning setup (Triantafillou et al., 2020).

Dataset We use the few-shot WSD benchmark provided by Holla et al. (2020a). It is based on the SemCor corpus (Miller et al., 1994), annotated with senses from the New Oxford American Dictionary by Yuan et al. (2016). The dataset consists of words grouped into meta-training, meta-validation and meta-test sets. The meta-test set consists of new words that were not part of meta-training and meta-validation sets. There are four setups varying in the number of sentences in the support set $|S| = 4, 8, 16, 32$. $|S| = 4$ corresponds to an extreme few-shot learning scenario for most words, whereas $|S| = 32$ comes closer to the number of sentences per word encountered in standard WSD setups. For $|S| = 4, 8, 16, 32$, the number of unique words in the meta-training / meta-validation / meta-test sets is 985/166/270, 985/163/259, 799/146/197 and 580/85/129 respectively. We use the publicly available standard dataset splits.[2]

Episodes The meta-training episodes were created by first sampling a set of words and a fixed number of senses per word, followed by sampling example sentences for these word-sense pairs. This strategy allows for a combinatorially large number of episodes. Every meta-training episode has $|S|$ sentences in both the support and query sets, and corresponds to the distinct task of disambiguating between the sampled word-sense pairs. The total number of meta-training episodes is $10,000$. In the meta-validation and meta-test sets, each episode corresponds to the task of disambiguating a single, previously unseen word between all its senses. For every meta-test episode, the model is fine-tuned on a few examples in the support set and its generalizability is evaluated on the query set. In contrast to the meta-training episodes, the meta-test episodes reflect a natural distribution of senses in the corpus, including class imbalance, providing a realistic evaluation setting.

[2] https://github.com/Nithin-Holla/MetaWSD

4 Methods

4.1 Model architectures

We experiment with the same model architectures as Holla et al. (2020a). The model f_θ, with parameters θ, takes words $\mathbf{x}_i$ as input and produces a per-word representation vector $f_\theta(\mathbf{x}_i)$ for $i = 1, ..., L$ where L is the length of the sentence. Sense predictions are only made for ambiguous words using the corresponding word representation.

GloVe+GRU Single-layer bi-directional GRU (Cho et al., 2014) network followed by a single linear layer, that takes GloVe embeddings (Pennington et al., 2014) as input. GloVe embeddings capture all senses of a word. We thus evaluate a model's ability to disambiguate from sense-agnostic input.

ELMo+MLP A multi-layer perception (MLP) network that receives contextualized ELMo embeddings (Peters et al., 2018) as input. Their contextualised nature makes ELMo embeddings better suited to capture meaning variation than the static ones. Since ELMo is not fine-tuned, this model has the lowest number of learnable parameters.

BERT Pretrained $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019) model followed by a linear layer, fully fine-tuned on the task. BERT underlies state-of-the-art approaches to WSD.

4.2 Prototypical Network

Our few-shot learning approach builds upon prototypical networks (Snell et al., 2017), which is widely used for few-shot image classification and has been shown to be successful in WSD (Holla et al., 2020a). It computes a prototype $\mathbf{z}_k = \frac{1}{K} \sum_k f_\theta(\mathbf{x}_k)$ of each word sense (where K is the number of examples for each word sense) through an embedding function f_θ, which is realized as the aforementioned architectures. It computes a distribution over classes for a query sample $\mathbf{x}$ given a distance function $d(\cdot, \cdot)$ as the softmax over its distances to the prototypes in the embedding space:

$$p(\mathbf{y}_i = k|\mathbf{x}) = \frac{\exp(-d(f_\theta(\mathbf{x}), \mathbf{z}_k))}{\sum_{k'} \exp(-d(f_\theta(\mathbf{x}), \mathbf{z}_{k'}))} \quad (1)$$

However, the resulting prototypes may not be sufficiently representative of word senses as semantic categories when using a single deterministic vector, computed as the average of only a few examples. Such representations lack expressiveness and may not encompass sufficient intra-class variance, that is needed to distinguish between different fine-grained word senses. Moreover, large uncertainty arises in the single prototype due to the small number of samples.

4.3 Variational Prototype Network

Variational prototype network (Zhen et al., 2020) (VPN) is a powerful model for learning latent representations from small amounts of data, where the prototype $\mathbf{z}$ of each class is treated as a distribution. Given a task with a support set S and query set Q, the objective of VPN takes the following form:

$$\mathcal{L}_{\text{VPN}} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\Big[\frac{1}{L_{\mathbf{z}}}\sum_{l_{\mathbf{z}}=1}^{L_{\mathbf{z}}} - \log p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{z}^{(l_{\mathbf{z}})}) + \lambda D_{\text{KL}}[q(\mathbf{z}|S)||p(\mathbf{z}|\mathbf{x}_i)]\Big] \tag{2}$$

where $q(\mathbf{z}|S)$ is the variational posterior over $\mathbf{z}$, $p(\mathbf{z}|\mathbf{x}_i)$ is the prior, and $L_{\mathbf{z}}$ is the number of Monte Carlo samples for $\mathbf{z}$. The prior and posterior are assumed to be Gaussian. The re-parameterization trick (Kingma and Welling, 2013) is adopted to enable back-propagation with gradient descent, i.e., $\mathbf{z}^{(l_{\mathbf{z}})} = f(S, \epsilon^{(l_{\mathbf{z}})})$, $\epsilon^{(l_{\mathbf{z}})} \sim \mathcal{N}(0, I)$, $f(\cdot,\cdot) = \epsilon^{(l_{\mathbf{z}})} * \mu_z + \sigma_z$, where the mean μ_z and diagonal covariance σ_z are generated from the posterior inference network with S as input. The amortization technique is employed for the implementation of VPN. The posterior network takes the mean word representations in the support set S as input and returns the parameters of $q(\mathbf{z}|S)$. Similarly, the prior network produces the parameters of $p(\mathbf{z}|\mathbf{x}_i)$ by taking the query word representation $\mathbf{x_i} \in \mathcal{Q}$ as input. The conditional predictive log-likelihood is implemented as a cross-entropy loss.

4.4 β-Variational Semantic Memory

In order to leverage the shared common knowledge between different tasks to improve disambiguation in future tasks, we incorporate variational semantic memory (VSM) as in Zhen et al. (2020). It consists of two main processes: *memory recall*, which retrieves relevant information that fits with specific tasks based on the support set of the current task;

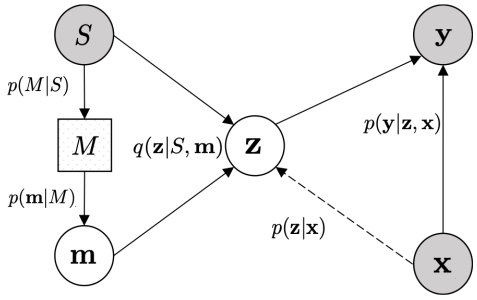


Figure 1: Computational graph of variational semantic memory for few-shot WSD. M is the semantic memory module, S the support set, $\mathbf{x}$ and $\mathbf{y}$ are the query sample and label, and $\mathbf{z}$ is the word sense prototype.

memory update, which effectively collects new information from the task and gradually consolidates the semantic knowledge in the memory. We adopt a similar memory mechanism and introduce an improved update rule for memory consolidation.

Memory recall The memory recall of VSM aims to choose the related content from the memory, and is accomplished by variational inference. It introduces latent memory $\mathbf{m}$ as an intermediate stochastic variable, and infers $\mathbf{m}$ from the addressed memory M. The approximate variational posterior $q(\mathbf{m}|M, S)$ over the latent memory $\mathbf{m}$ is obtained empirically by

$$q(\mathbf{m}|M, S) = \sum_{a=1}^{|M|}\gamma_a p(\mathbf{m}|M_a), \tag{3}$$

where

$$\gamma_a = \frac{\exp\big(g(M_a, S)\big)}{\sum_i \exp\big(g(M_i, S)\big)} \tag{4}$$

$g(\cdot)$ is the dot product, $|M|$ is the number of memory slots, M_a is the memory content at slot a and stores the prototype of samples in each class, and we take the mean representation of samples in S.

The variational posterior over the prototype then becomes:

$$\tilde{q}(\mathbf{z}|M, S) \approx \frac{1}{L_{\mathbf{m}}}\sum_{l_{\mathbf{m}}=1}^{L_m} q(\mathbf{z}|\mathbf{m}^{(l_{\mathbf{m}})}, S), \tag{5}$$

where $\mathbf{m}^{(l_{\mathbf{m}})}$ is a Monte Carlo sample drawn from the distribution $q(\mathbf{m}|M, S)$, and $l_{\mathbf{m}}$ is the number of samples. By incorporating the latent memory $\mathbf{m}$ from Eq. (3), we achieve the objective for varia-

tional semantic memory as follows:

$$\mathcal{L}_{\text{VSM}} = \sum_{i=1}^{|Q|} \Big[- \mathbb{E}_{q(\mathbf{z}|S,\mathbf{m})} \big[\log p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{z}) \big] + \lambda_{\mathbf{z}} D_{\text{KL}} \big[q(\mathbf{z}|S,\mathbf{m}) || p(\mathbf{z}|\mathbf{x}_i) \big] + \lambda_{\mathbf{m}} D_{\text{KL}} \big[\sum_{i}^{|M|} \gamma_i p(\mathbf{m}|M_i) || p(\mathbf{m}|S) \big] \Big] \tag{6}$$

where $p(\mathbf{m}|S)$ is the introduced prior over $\mathbf{m}$, $\lambda_{\mathbf{z}}$ and $\lambda_{\mathbf{m}}$ are the hyperparameters. The overall computational graph of VSM is shown in Figure 1. Similarly, the posterior and prior over $\mathbf{m}$ are also assumed to be Gaussian and obtained by using amortized inference networks; more details are provided in Appendix A.1.

Memory update The memory update is to be able to effectively absorb new useful information to enrich memory content. VSM employs an update rule as follows:

$$M_c \leftarrow \beta M_c + (1-\beta)\bar{M}_c, \tag{7}$$

where M_c is the memory content corresponding to class c, $\bar{M}_c$ is obtained using graph attention (Veličković et al., 2017), and $\beta \in (0, 1)$ is a hyperparameter.

Adaptive memory update Although VSM was shown to be promising for few-shot image classification, it can be seen from the experiments by Zhen et al. (2020) that different values of β have considerable influence on the performance. β determines the extent to which memory is updated at each iteration. In the original VSM, β is treated as a hyperparameter obtained by cross-validation, which is time-consuming and inflexible in dealing with different datasets. To address this problem, we propose an adaptive memory update rule by learning β from data using a lightweight hypernetwork (Ha et al., 2016). To be more specific, we obtain β by a function $f_\beta(\cdot)$ implemented as an MLP with a sigmoid activation function in the output layer. The hypernetwork takes $\bar{M}_c$ as input and returns the value of β:

$$\beta = f_\beta(\bar{M}_c) \tag{8}$$

Moreover, to prevent the possibility of endless growth of memory value, we propose to scale down the memory value whenever $\|M_c\|_2 > 1$. This is achieved by scaling as follows:

$$M_c = \frac{M_c}{\max(1, \|M_c\|_2)} \tag{9}$$

When we update memory, we feed the new obtained memory $\bar{M}_c$ into the hypernetwork $f_\beta(\cdot)$ and output adaptive β for the update. We provide a more detailed implementation of β-VSM in Appendix A.1.

5 Experiments and results

Experimental setup The size of the shared linear layer and memory content of each word sense is 64, 256, and 192 for GloVe+GRU, ELMo+MLP and BERT respectively. The activation function of the shared linear layer is tanh for GloVe+GRU and ReLU for the rest. The inference networks $g_\phi(\cdot)$ for calculating the prototype distribution and $g_\psi(\cdot)$ for calculating the memory distribution are all three-layer MLPs, with the size of each hidden layer being 64, 256, and 192 for GloVe+GRU, ELMo+MLP and BERT. The activation function of their hidden layers is ELU (Clevert et al., 2016), and the output layer does not use any activation function. Each batch during meta-training includes 16 tasks. The hypernetwork $f_\beta(\cdot)$ is also a three-layer MLP, with the size of hidden state consistent with that of the memory contents. The linear layer activation function is ReLU for the hypernetwork. For BERT and $|S| = \{4, 8\}$, $\lambda_{\mathbf{z}} = 0.001$, $\lambda_{\mathbf{m}} = 0.0001$ and learning rate is $5\text{e}{-6}$; $|S| = 16$, $\lambda_{\mathbf{z}} = 0.0001$, $\lambda_{\mathbf{m}} = 0.0001$ and learning rate is $1\text{e}{-6}$; $|S| = 32$, $\lambda_{\mathbf{z}} = 0.001$, $\lambda_{\mathbf{m}} = 0.0001$ and learning rate is $1\text{e}{-5}$. Hyperparameters for other models are reported in Appendix A.2. All the hyperparameters are chosen using the meta-validation set. The number of slots in memory is consistent with the number of senses in the meta-training set – 2915 for $|S| = 4$ and 8; 2452 for $|S| = 16$; 1937 for $|S| = 32$. The evaluation metric is the word-level macro F1 score, averaged over all episodes in the meta-test set. The parameters are optimized using Adam (Kingma and Ba, 2014).

We compare our methods against several baselines and state-of-the-art approaches. The nearest neighbor classifier baseline (*NearestNeighbor*) predicts a query example's sense as the sense of the support example closest in the word embedding space (ELMo and BERT) in terms of cosine distance. The episodic fine-tuning baseline (*EF-ProtoNet*) is one where only meta-testing is

Embedding/ Encoder	Method	Average macro F1 score $\|S\| = 4$	$\|S\| = 8$	$\|S\| = 16$	$\|S\| = 32$
-	MajoritySenseBaseline	0.247	0.259	0.264	0.261
GloVe+GRU	NearestNeighbor	–	–	–	–
	EF-ProtoNet	0.522 ± 0.008	0.539 ± 0.009	0.538 ± 0.003	0.562 ± 0.005
	ProtoNet	0.579 ± 0.004	0.601 ± 0.003	0.633 ± 0.008	0.654 ± 0.004
	ProtoFOMAML	0.577 ± 0.011	0.616 ± 0.005	0.626 ± 0.005	0.631 ± 0.008
	β-VSM (Ours)	$\mathbf{0.597 \pm 0.005}$	$\mathbf{0.631 \pm 0.004}$	$\mathbf{0.652 \pm 0.006}$	$\mathbf{0.678 \pm 0.007}$
ELMo+MLP	NearestNeighbor	0.624	0.641	0.645	0.654
	EF-ProtoNet	0.609 ± 0.008	0.635 ± 0.004	0.661 ± 0.004	0.683 ± 0.003
	ProtoNet	0.656 ± 0.006	0.688 ± 0.004	0.709 ± 0.006	0.731 ± 0.006
	ProtoFOMAML	0.670 ± 0.005	0.700 ± 0.004	0.724 ± 0.003	0.737 ± 0.007
	β-VSM (Ours)	$\mathbf{0.679 \pm 0.006}$	$\mathbf{0.709 \pm 0.005}$	$\mathbf{0.735 \pm 0.004}$	$\mathbf{0.758 \pm 0.005}$
BERT	NearestNeighbor	0.681	0.704	0.716	0.741
	EF-ProtoNet	0.594 ± 0.008	0.655 ± 0.004	0.682 ± 0.005	0.721 ± 0.009
	ProtoNet	0.696 ± 0.011	0.750 ± 0.008	0.755 ± 0.002	0.766 ± 0.003
	ProtoFOMAML	0.719 ± 0.005	0.756 ± 0.007	0.744 ± 0.007	0.761 ± 0.005
	β-VSM (Ours)	$\mathbf{0.728 \pm 0.012}$	$\mathbf{0.773 \pm 0.005}$	$\mathbf{0.776 \pm 0.003}$	$\mathbf{0.788 \pm 0.003}$

Table 1: Model performance comparison on the meta-test words using different embedding functions.

performed, starting from a randomly initialized model. Prototypical network (*ProtoNet*) and ProtoFOMAML achieve the highest few-shot WSD performance to date on the benchmark of Holla et al. (2020a).

Results In Table 1, we show the average macro F1 scores of the models, with their mean and standard deviation obtained over five independent runs. Our proposed β-VSM achieves the new state-of-the-art performance on few-shot WSD with all the embedding functions, across all the setups with varying $|S|$. For GloVe+GRU, where the input is sense-agnostic embeddings, our model improves disambiguation compared to ProtoNet by 1.8% for $|S| = 4$ and by 2.4% for $|S| = 32$. With contextual embeddings as input, β-VSM with ELMo+MLP also leads to improvements compared to the previous best ProtoFOMAML for all $|S|$. Holla et al. (2020a) obtained state-of-the-art performance with BERT, and β-VSM further advances this, resulting in a gain of $0.9 - 2.2\%$. The consistent improvements with different embedding functions and support set sizes suggest that our β-VSM is effective for few-shot WSD for varying number of shots and senses as well as across model architectures.

6 Analysis and discussion

To analyze the contributions of different components in our method, we perform an ablation study by comparing ProtoNet, VPN, VSM and β-VSM and present the macro F1 scores in Table 2.

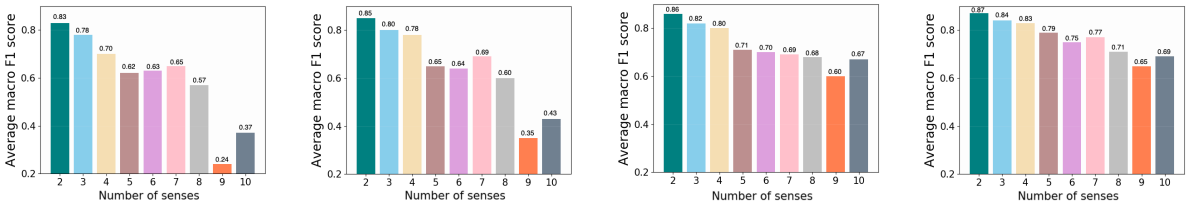
Role of variational prototypes VPN consistently outperforms ProtoNet with all embedding functions (by around 1% F1 score on average). The results indicate that the probabilistic prototypes provide more informative representations of word senses compared to deterministic vectors. The highest gains were obtained in case of GloVe+GRU (1.7% F1 score with $|S| = 8$), suggesting that probabilistic prototypes are particularly useful for models that rely on static word embeddings, as they capture uncertainty in contextual interpretation.

Role of variational semantic memory We show the benefit of VSM by comparing it with VPN. VSM consistently surpasses VPN with all three embedding functions. According to our analysis, VSM makes the prototypes of different word senses more distinctive and distant from each other. The senses in memory provide more context information, enabling larger intra-class variations to be captured, and thus lead to improvements upon VPN.

Role of adaptive β To demonstrate the effectiveness of the hypernetwork for adaptive β, we compare β-VSM with VSM where β is tuned by cross-validation. It can be seen from Table 2 that there is consistent improvement over VSM. Thus, the learned adaptive β acquires the ability to determine how much of the contents of memory needs to be updated based on the current new memory. β-VSM enables the memory content of different word senses to be more representative by better absorbing information from data with adaptive update, resulting in improved performance.

Embedding/ Encoder	Method	Average macro F1 score $\|S\| = 4$	$\|S\| = 8$	$\|S\| = 16$	$\|S\| = 32$
GloVe+GRU	ProtoNet	0.579 ± 0.004	0.601 ± 0.003	0.633 ± 0.008	0.654 ± 0.004
	VPN	0.583 ± 0.005	0.618 ± 0.005	0.641 ± 0.007	0.668 ± 0.005
	VSM	0.587 ± 0.004	0.625 ± 0.004	0.645 ± 0.006	0.670 ± 0.005
	β-VSM	**0.597 ± 0.005**	**0.631 ± 0.004**	**0.652 ± 0.006**	**0.678 ± 0.007**
ELMo+MLP	ProtoNet	0.656 ± 0.006	0.688 ± 0.004	0.709 ± 0.006	0.731 ± 0.006
	VPN	0.661 ± 0.005	0.694 ± 0.006	0.718 ± 0.004	0.741 ± 0.004
	VSM	0.670 ± 0.006	0.707 ± 0.006	0.726 ± 0.005	0.750 ± 0.004
	β-VSM	**0.679 ± 0.006**	**0.709 ± 0.005**	**0.735 ± 0.004**	**0.758 ± 0.005**
BERT	ProtoNet	0.696 ± 0.011	0.750 ± 0.008	0.755 ± 0.002	0.766 ± 0.003
	VPN	0.703 ± 0.011	0.761 ± 0.007	0.762 ± 0.004	0.779 ± 0.002
	VSM	0.717 ± 0.013	0.769 ± 0.006	0.770 ± 0.005	0.784 ± 0.002
	β-VSM	**0.728 ± 0.012**	**0.773 ± 0.005**	**0.776 ± 0.003**	**0.788 ± 0.003**

Table 2: Ablation study comparing the meta-test performance of the different variants of prototypical networks.

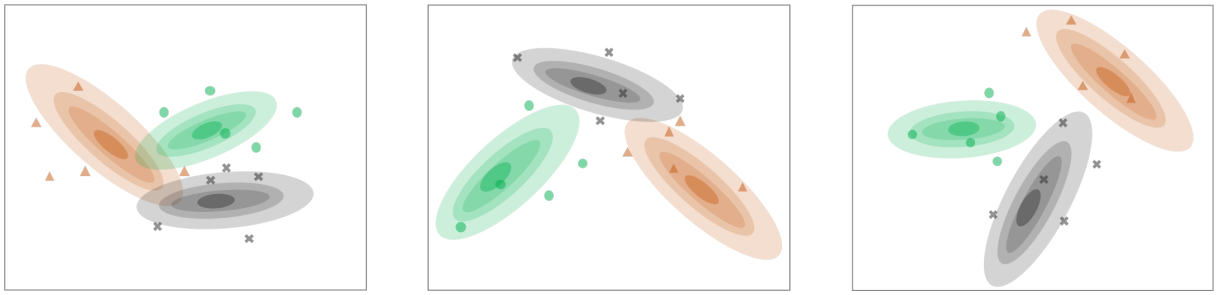


(a) ProtoNet (b) VPN (c) VSM (d) β-VSM

Figure 2: Distribution of average macro F1 scores over number of senses for BERT-based models with $|S| = 16$.

Variation of performance with the number of senses In order to further probe into the strengths of β-VSM, we analyze the macro F1 scores of the different models averaged over all the words in the meta-test set with a particular number of senses. In Figure 2, we show a bar plot of the scores obtained from BERT for $|S| = 16$. For words with a low number of senses, the task corresponds to a higher number of effective shots and vice versa. It can be seen that the different models perform roughly the same for words with fewer senses, i.e., 2 – 4. VPN is comparable to ProtoNet in its distribution of scores. But with semantic memory, VSM improves the performance on words with a higher number of senses. β-VSM further boosts the scores for such words on average. The same trend is observed for $|S| = 8$ (see Appendix A.3). Therefore, the improvements of β-VSM over ProtoNet come from tasks with fewer shots, indicating that VSM is particularly effective at disambiguation in low-shot scenarios.

Visualization of prototypes To study the distinction between the prototype distributions of word senses obtained by β-VSM, VSM and VPN, we visualize them using t-SNE (Van der Maaten and Hinton, 2008). Figure 3 shows prototype distributions based on BERT for the word *draw*. Different colored ellipses indicate the distribution of its different senses obtained from the support set. Different colored points indicate the representations of the query examples. β-VSM makes the prototypes of different word senses of the same word more distinctive and distant from each other, with less overlap, compared to the other models. Notably, the representations of query examples are closer to their corresponding prototype distribution for β-VSM, thereby resulting in improved performance. We also visualize the prototype distributions of similar vs. dissimilar senses of multiple words in Figure 4 (see Appendix A.4 for example sentences). The blue ellipse corresponds to the 'set up' sense of *launch* from the meta-test samples. Green and gray ellipses correspond to a similar sense of the words *start* and *establish* from the memory. We can see that they are close to each other. Orange and purple ellipses correspond to other senses of the words *start* and *establish* from the memory, and they are well separated. For a given query word, our model is thus able to retrieve related senses from the memory and exploit them to make its word sense distribution more representative and distinctive.



(a) VPN (b) VSM (c) β-VSM

Figure 3: Prototype distributions of distinct senses of *draw* with different models.

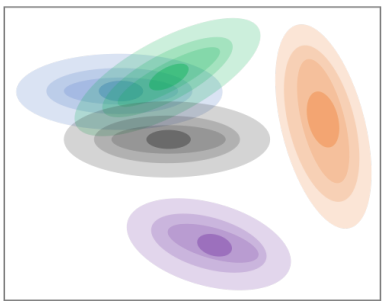


Figure 4: Prototype distributions of similar sense of *launch* (blue), *start* (green) and *establish* (grey). Distinct senses: *start* (orange) and *establish* (purple).

7 Conclusion

In this paper, we presented a model of variational semantic memory for few-shot WSD. We use a variational prototype network to model the prototype of each word sense as a distribution. To leverage the shared common knowledge between tasks, we incorporate semantic memory into the probabilistic model of prototypes in a hierarchical Bayesian framework. VSM is able to acquire long-term, general knowledge that enables learning new senses from very few examples. Furthermore, we propose adaptive β-VSM which learns an adaptive memory update rule from data using a lightweight hypernetwork. The consistent new state-of-the-art performance with three different embedding functions shows the benefit of our model in boosting few-shot WSD.

Since meaning disambiguation is central to many natural language understanding tasks, models based on semantic memory are a promising direction in NLP, more generally. Future work might investigate the role of memory in modeling meaning variation across domains and languages, as well as in tasks that integrate knowledge at different levels of linguistic hierarchy.

References

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

Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2019. How to train your MAML. In *International Conference on Learning Representations*.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2019. Learning to few-shot learn across diverse natural language classification tasks. *arXiv preprint arXiv:1911.03863*.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 522–534, Online. Association for Computational Linguistics.

Y. Bengio, S. Bengio, and J. Cloutier. 1991. Learning a synaptic learning rule. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume ii, pages 969 vol.2–.

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.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *4th*

International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365, Vancouver, Canada. Association for Computational Linguistics.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32*, pages 13143–13152. Curran Associates, Inc.

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.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. 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 1192–1197, Hong Kong, China. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia. PMLR.

Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1087–1094, Online. Association for Computational Linguistics.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. 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 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

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.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6440, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020a. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4517–4533, Online. Association for Computational Linguistics.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020b. Meta-learning with sparse experience replay for lifelong language learning. *arXiv preprint arXiv:2009.04891*.

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 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 897–907, Berlin, Germany. Association for Computational Linguistics.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Dmitry Krotov and John J Hopfield. 2016. Dense associative memory for pattern recognition. *arXiv preprint arXiv:1606.01164*.

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.

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

Zheng Li, Mukul Kumar, William Headden, Bing Yin, Ying Wei, Yu Zhang, and Qiang Yang. 2020. Learn to cross-lingual transfer with meta graph learning across heterogeneous languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2290–2301, Online. Association for Computational Linguistics.

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

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

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

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

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

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.

Tsendsuren Munkhdalai, Alessandro Sordoni, Tong Wang, and Adam Trischler. 2019. Metalearned neural memory. In *Advanced in Neural Information Processing Systems*.

Tsendsuren Munkhdalai and Hong Yu. 2017a. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2554–2563, International Convention Centre, Sydney, Australia. PMLR.

Tsendsuren Munkhdalai and Hong Yu. 2017b. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2554–2563, International Convention Centre, Sydney, Australia. PMLR.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. 2018. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR.

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

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Abiola Obamuyide and Andreas Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5873–5879, Florence, Italy. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, Florence, Italy. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

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

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Eleanor Rosch. 1975. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:192–233.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30*, pages 3856–3866.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016a. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016b. Meta-learning with memory-augmented neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1842–1850, New York, New York, USA. PMLR.

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.

Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, pages 4077–4087.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224, Austin, Texas. Association for Computational Linguistics.

Sebastian Thrun and Lorien Pratt, editors. 1998. *Learning to Learn*. Kluwer Academic Publishers, USA.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. 2020. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pages 3630–3638.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2692–2700. Curran Associates, Inc.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Changlong Yu, Jialong Han, Haisong Zhang, and Wilfred Ng. 2020. Hypernymy detection for low-resource languages via meta learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3656, Online. Association for Computational Linguistics.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. 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 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

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

Xiantong Zhen, Yingjun Du, Huan Xiong, Qiang Qiu, Cees Snoek, and Ling Shao. 2020. Learning to learn variational semantic memory. In *Proceedings of NeurIPS*.

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

A Appendix

A.1 Implementation details

In the meta-training phase, we implement β-VSM by end-to-end learning with stochastic neural networks. The inference network and hypernetwork are parameterized by a feed-forward multi-layer perceptrons (MLP). At meta-train time, we first extract the features of the support set via $f_\theta(\mathbf{x}_S)$, where f_θ is the feature extraction network and we use permutation-invariant instance-pooling operations to get the mean feature $\bar{f}_c^s$ of samples in the c-th class. Then we get the memory M_a by using the support representation $\bar{f}_c^s$ of each class. The memory obtained M_a will be fed into a small three-layers MLP network $g_\psi(\cdot)$ to calculate the mean $\boldsymbol{\mu}_\mathbf{m}$ and variance $\boldsymbol{\sigma}_\mathbf{m}$ of the memory distribution $\mathbf{m}$, which is then used to sample the memory $\mathbf{m}$ by $\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{m}, diag((\boldsymbol{\sigma}_\mathbf{m})^2))$. The new memory $\bar{M}_c$ is obtained by using graph attention. The nodes of the graph are a set of feature representations of the current task samples: $F_c = \{f_c^0, f_c^1, f_c^2, \ldots, f_c^{\mathcal{N}_c}\}$, where $f_c^{\mathcal{N}_c} \in \mathbb{R}^d$, $\mathcal{N}_c = |S_c \cup Q_c|$, $f_c^0 = M_c$, $f_c^{i>0} = f_\theta(\mathbf{x}_c^i)$. $\mathcal{N}_c$ contains all samples including both the support and query set from the c-th category in the current task. When we update memory, we take the new obtained memory $\bar{M}_c$ into the hypernetwork $f_\beta(\cdot)$ as input and output the adaptive β to update the memory using Equation 8. We calculate the prototype of the latent distribution, i.e., the mean $\boldsymbol{\mu}_\mathbf{z}$ and variance $\boldsymbol{\sigma}_\mathbf{z}$ by another small three-layer MLP network $g_\phi(\cdot,\cdot)$, whose inputs are $\bar{f}_c^s$ and $\mathbf{m}$. Then the prototype $\mathbf{z}^{(l_\mathbf{z})}$ is sampled from the distribution $\mathbf{z}^{(l_\mathbf{z})} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, diag((\boldsymbol{\sigma}_\mathbf{z})^2))$. By using the prototypical word sense of support samples and the feature embedding of query sample $\mathbf{x}_i$, we obtain the predictive value $\hat{\mathbf{y}}_i$.

At meta-test time, we feed the support representation $\bar{f}_c^s$ into the $g_\psi(\cdot)$ to generate the memory $\mathbf{m}_a$. Then, using the sampled memory $\mathbf{m}_a$ and the support representation $\bar{f}_c^s$, we obtain the distribution of prototypical word sense $\mathbf{z}$. Finally, we make predictions for the query sample by using the query representation extracted from embedding function and the support prototype $\mathbf{z}$.

A.2 Hyperparameters and runtimes

We present our hyperparameters in Table 3. For Monte Carlo sampling, we set different L_Z and L_M for the each embedding function and $|S|$, which are chosen using the validation set. Training time differs for different $|S|$ and different embedding functions. Here we give the training time per epoch for $|S| = 16$. For GloVe+GRU, the approximate training time per epoch is 20 minutes; for ELMo+MLP it is 80 minutes; and for BERT, it is 60 minutes. The number of meta-learned parameters for GloVe+GRU is θ are $889,920$; for ELMo+MLP it is $262,404$; and for BERT it is θ are $107,867,328$. We implemented all models using the PyTorch framework and trained them on an NVIDIA Tesla V100.

Embedding/ Encoder	$\|S\|$	Learning rate	λ_z	λ_m	$L_\ddagger$	$L_{\mathbf{M}}$
GloVe+GRU	4	1e−5	1e−3	1e−4	200	150
	8	1e−5	1e−3	1e−4	200	150
	16	1e−4	1e−4	1e−3	150	150
	32	1e−4	1e−3	1e−3	150	150
ELMo+MLP	4	1e−5	1e−4	1e−4	200	150
	8	1e−5	1e−4	1e−4	200	150
	16	1e−4	1e−3	1e−3	150	150
	32	1e−4	1e−3	1e−3	150	150
BERT	4	5e−6	1e−3	1e−4	200	200
	8	5e−6	1e−3	1e−4	200	200
	16	1e−6	1e−4	1e−4	150	150
	32	1e−4	1e−3	1e−4	150	100

Table 3: Hyperparameters used for training the models.

A.3 Variation of performance with the number of senses

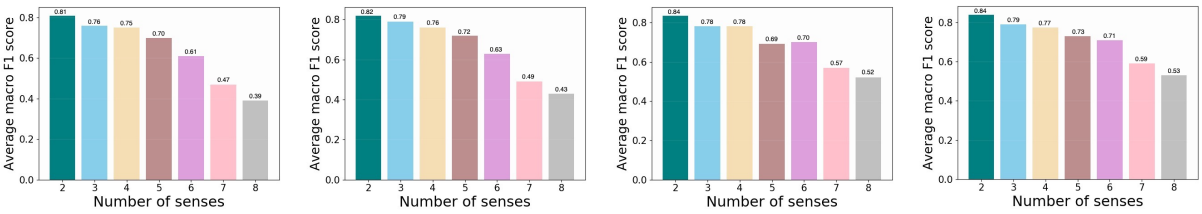
To further demonstrate that β-VSM achieves better performance in extremely data scarce scenarios, we also analyze variation of macro F1 scores with the number of senses for BERT and $|S| = 8$. In Figure 5, we observe a similar trend as with $|S| = 16$. β-VSM has an improved performance for words with many senses, which corresponds to a low-shot scenario. For example, with 8 senses, the task is essentially one-shot.

A.4 Example sentences to visualize prototypes

In Table 4, we provide some example sentences used to generate the plots in Figure 4. These examples correspond to words *launch*, *start* and *establish*, and contain senses 'set up', 'begin' and 'build up'.

A.5 Results on the meta-validation set

We provide the results on the on the meta-validation set in the Table 5, to better facilitate reproducibility.



(a) ProtoNet (b) VPN (c) VSM (d) β-VSM

Figure 5: Distribution of average macro F1 scores over number of senses for BERT-based models with $|S| = 8$.

Word	**Sense**	**Sentence**
launch	set up	The Corinthian Yacht Club in Tiburon launches its winter races Nov. 5.
launch	set up	The most infamous of all was launched by the explosion of the island of Krakatoa in 1883; it raced across the Pacific at 300 miles an hour devastated the coasts of Java and Sumatra with waves 100 to 130 feet high, and pounded the shore as far away as San Francisco.
launch	set up	In several significant cases, such as India, a decade of concentrated effort can launch these countries into a stage in which they can carry forward their own economic and social progress with little or no government-to-government assistance.
start	set up	With these maps completed, the inventory phase of the plan has been started.
start	begin	Congress starts another week tomorrow with sharply contrasting forecasts for the two chambers.
establish	set up	For the convenience of guests bundle centers have been established throughout the city and suburbs where the donations may be deposited between now and the date of the big event.
establish	build up	From the outset of his first term, he established himself as one of the guiding spirits of the House of Delegates.

Table 4: Example sentences for different word-sense pairs used to generate the visualization in Figure 4.

Embedding/ Encoder	**Method**	**Average macro F1 score** $\|S\| = 4$	$\|S\| = 8$	$\|S\| = 16$	$\|S\| = 32$
GloVe+GRU	ProtoNet	0.591 ± 0.008	0.615 ± 0.001	0.638 ± 0.007	0.634 ± 0.006
	VPN	0.602 ± 0.004	0.624 ± 0.004	0.646 ± 0.006	0.651 ± 0.005
	VSM	0.617 ± 0.005	0.635 ± 0.005	0.649 ± 0.004	0.673 ± 0.006
	β-VSM	$\mathbf{0.622 \pm 0.005}$	$\mathbf{0.649 \pm 0.004}$	$\mathbf{0.657 \pm 0.005}$	$\mathbf{0.680 \pm 0.006}$
ELMo+MLP	ProtoNet	0.682 ± 0.008	0.701 ± 0.007	0.741 ± 0.007	0.722 ± 0.011
	VPN	0.689 ± 0.004	0.709 ± 0.006	0.749 ± 0.005	0.748 ± 0.004
	VSM	0.693 ± 0.005	0.712 ± 0.007	0.754 ± 0.006	0.755 ± 0.006
	β-VSM	$\mathbf{0.701 \pm 0.006}$	$\mathbf{0.723 \pm 0.005}$	$\mathbf{0.760 \pm 0.005}$	$\mathbf{0.761 \pm 0.004}$
BERT	ProtoNet	0.742 ± 0.007	0.759 ± 0.013	0.786 ± 0.004	0.770 ± 0.009
	VPN	0.752 ± 0.011	0.769 ± 0.005	0.793 ± 0.003	0.785 ± 0.004
	VSM	0.767 ± 0.009	0.778 ± 0.005	0.801 ± 0.006	0.815 ± 0.005
	β-VSM	$\mathbf{0.771 \pm 0.008}$	$\mathbf{0.784 \pm 0.006}$	$\mathbf{0.810 \pm 0.004}$	$\mathbf{0.829 \pm 0.004}$

Table 5: Average macro F1 scores of the meta-validation words.