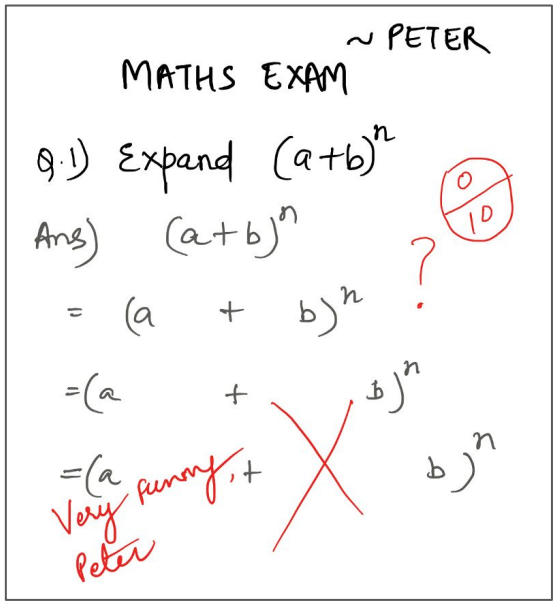
Word Sense Disambiguation


~ PETER
MATHS EXAM
Q.1) Expand $(a+b)^n$
Ans) $(a+b)^n$
$= (a + b)^n$
$= (a + b)^n$
$= (a + b)^n$
Very funny, Peter
0/10
?

Why disambiguation is needed?

Human language is often ambiguous.

Word can have multiple meaning - polysemy

- Bar, bank, plot, ...

Word can have same sound - homonyms

- May, sun, ...

Words can have different part of speech - POS ambiguity

- Drive, round, ….

Applications

- Machine Translation

  - Translate “bill” from English to Spanish

    Is it a “pico” or a “cuenta”?

    Is it a bird jaw or an invoice?

- Information Retrieval

  - Find all webpages about “bat”

    The sport accessory or the animal?

- Question Answering

  - “A far cry” was written by Winston Churchill.

    The Novelist or British politician?

- Knowledge Acquisition

  - Add to Knowledge base: Anne Hidalgo is the mayor of Paris.

    France or Texas?

Defining WSD

Given a target word and context we have to ***determine the sense the word corresponds to***.

Given a pre-defined inventory of word senses

- **Given a text**
- **Tag each ambiguous word occurrence with the most likely word sense.**

How do humans disambiguate?

“Big rig carrying fruit crashes on 210 Freeway, creates jam”

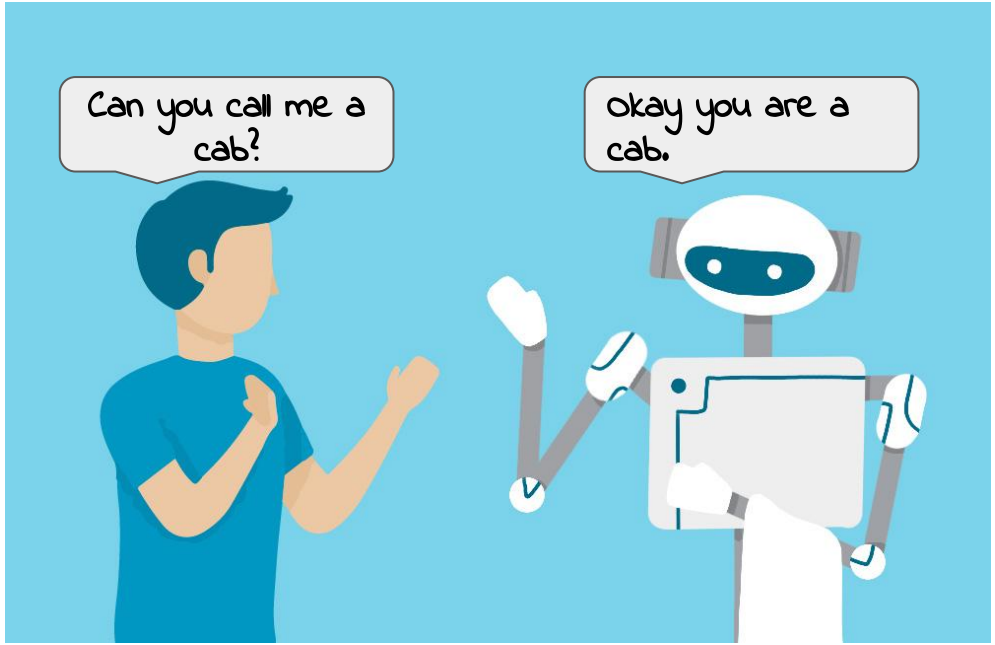
How do humans disambiguate?

“Big rig carrying fruit crashes on 210 Freeway, creates jam”

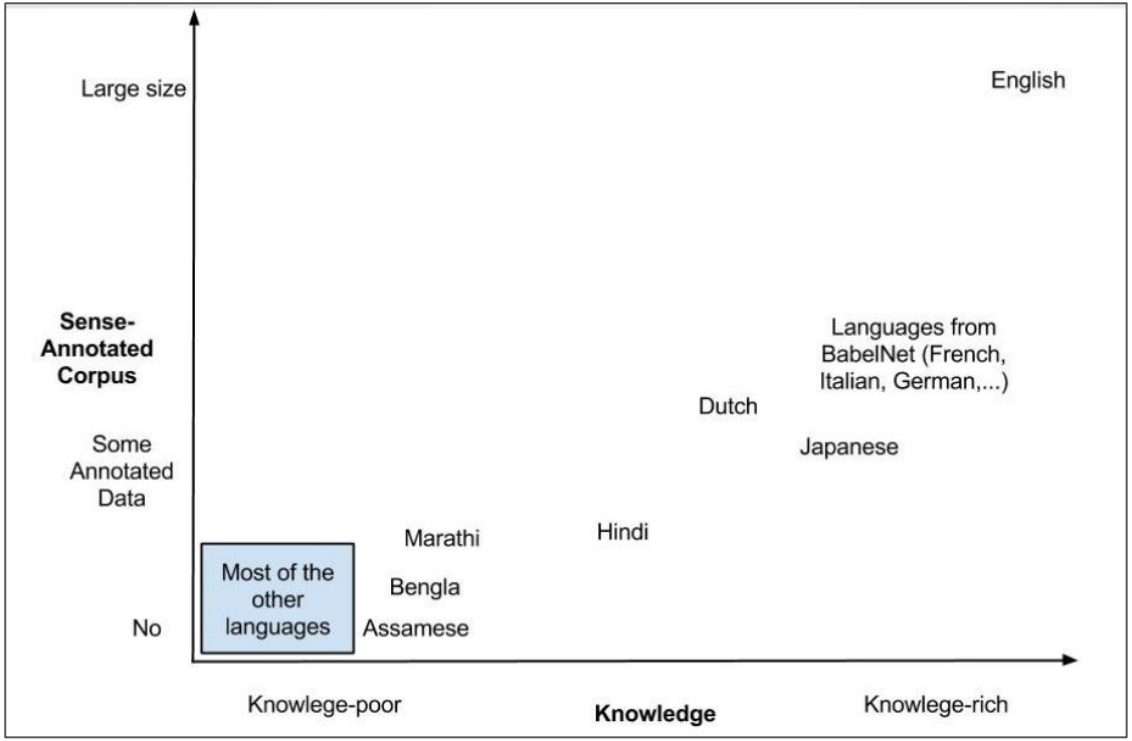
>>> Context, common sense knowledge

Teaching machines to disambiguate

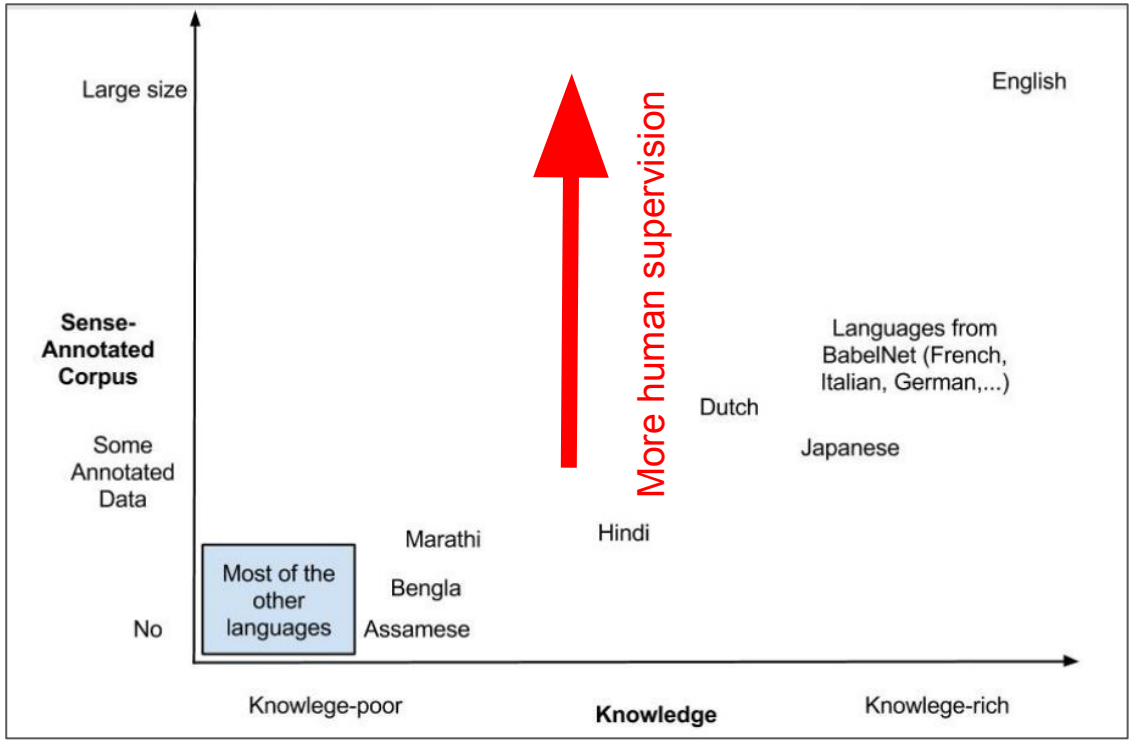
- Using similarity
- One classifier for all.
- One classifier for each word.
- So on...



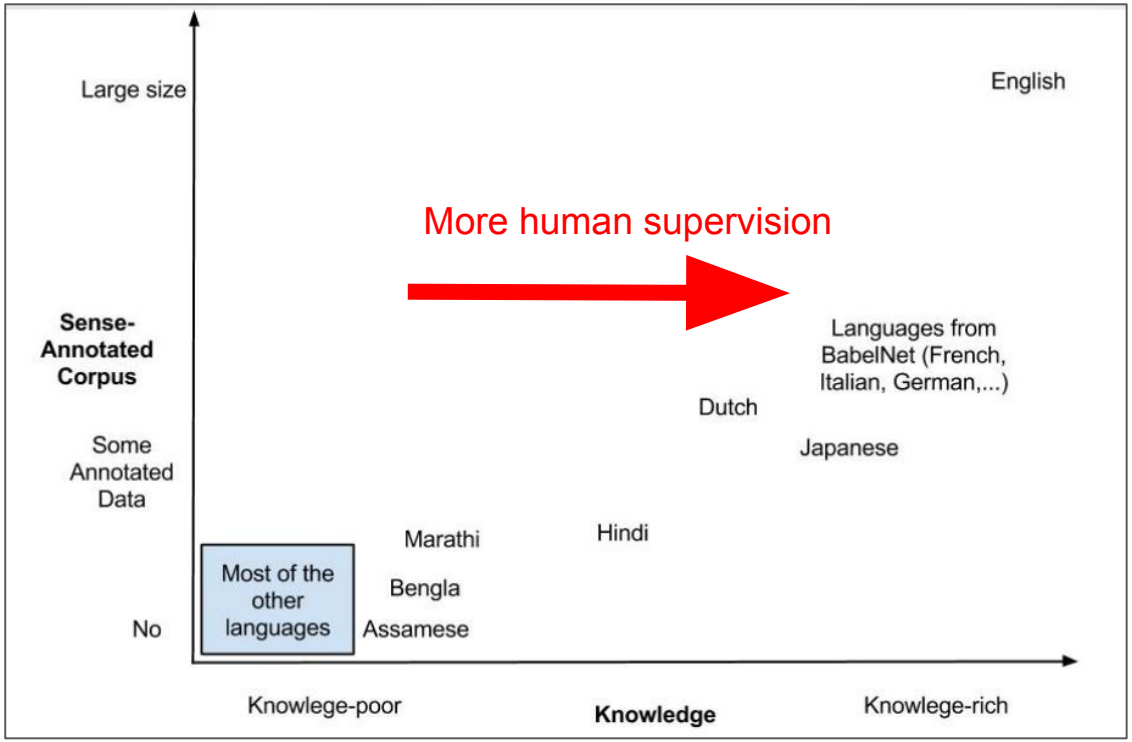
Teaching machines to disambiguate



Teaching machines to disambiguate



Teaching machines to disambiguate



Works	Methods
Lee and Ng et al 2002, Navigli et al 2009	Hand crafted features from context words
Taghipour and Ng et al 2015	Word embeddings for features
Raganato et al 2017b	Training individual models per word
Peters et al 2018	Single LSTM model with all classes
Peters et al 2018	Nearest neighbour with Elmo embeddings
Hadiwinoto et al 2019	Pre-trained contextualized embeddings from BERT as features
Huang et al 2019	Incorporating sense definition from Wordnet using context-gloss paris to fine-tune BERT
Bevilacqua et al 2020	Using wordnet's graph structure

Different words have different set of classes.

Variance of number of examples for each classes give rise to high class imbalance.

A new approach to handle such scenario of (N classes, K examples) has been introduced recently - Meta learning.

Learning to Learn to Disambiguate: Meta-Learning for Few-Shot Word Sense Disambiguation

Nithin Holla
ILLC, University of Amsterdam
nithin.holla7@gmail.com

Pushkar Mishra
Facebook AI
pushkarmishra@fb.com

Helen Yannakoudakis
Dept. of Informatics, King's College London
helen.yannakoudakis@kcl.ac.uk

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

Abstract

The success of deep learning methods hinges on the availability of large training datasets annotated for the task of interest. In contrast to human intelligence, these methods lack versatility and struggle to learn and adapt quickly to new tasks, where labeled data is scarce. Meta-learning aims to solve this problem by training a model on a large number of few-shot tasks, with an objective to learn new tasks quickly from a small number of examples. In this paper, we propose a meta-learning framework for few-shot word sense disambiguation (WSD), where the goal is to learn to disambiguate un-

2016; Yuan et al., 2016) or are knowledge-based (Lesk, 1986; Agirre et al., 2014; Moro et al., 2014). While supervised methods generally outperform the knowledge-based ones (Raganato et al., 2017a), they require data manually annotated with word senses, which are expensive to produce at a large scale. These methods also tend to learn a classification model for each word independently, and hence may perform poorly on words that have a limited amount of annotated data. Yet, alternatives that involve a single supervised model for all words (Raganato et al., 2017b) still do not adequately solve the problem for rare words (Kumar et al., 2019).

Meta learning is based on non- traditional learning paradigm of episodic learning.

Episodic learning refers to learning from episodes insides of data points. These episodes consists of a task T_i comprising a small number of examples - D_support_i and D_evaluation_i.

A typical setup for meta-learning is the balanced N-way,K-shot setting where each episode has N classes with K examples per class in its support set.

Meta learning algorithms

- Metric based - embed examples of episodes into high dim spaces using nn and then use it obtain probability distribution over the query examples (Koch et al 2015; Snell et al 2017)

- Model based - aims to achieve learning directly through their architecture - they employ external memory to learn (Santoro et al 2016)

- Optimization based - includes explicitly generalizability in their objective function. (Ravi and Larochelle 2017, Finn et al 2017)

Prototypical Network (Proto) *(metric based)*

- Embedding network f_theta to obtain prototype vector for every class from the support set.

$$\boldsymbol{\mu_c} = \frac{1}{|S_c|} \sum_{\boldsymbol{x}_i \in S_c} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

$$\boldsymbol{\theta} \leftarrow \text{Adam}(\mathcal{L}^q_{\mathcal{T}_i}, \boldsymbol{\theta}, \beta)$$

Model Agnostic Meta learning (MAML) *(optimization based)*

- It is designed for N-way, K-shot classification task.

Optimization goal : is to train a model's initial parameters such that it can perform well on a new task after only a few gradient steps on a small amount of data. Tasks are drawn from a distribution p(T).

$$\boldsymbol{\theta}_i' = U(\mathcal{L}^s_{\mathcal{T}_i}, \boldsymbol{\theta}, \alpha, m),$$

The meta-objective is to have $f_{\theta' i}$ generalize well across tasks from p(T):

$$J(\boldsymbol{\theta}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}^q_{\mathcal{T}_i}(f_{U(\mathcal{L}^s_{\mathcal{T}_i}, \boldsymbol{\theta}, \alpha, m)}).$$

Model Agnostic Meta learning (MAML) *(optimization based)*

- outer-loop optimization,does the update with the outer-loop learning rate β

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}^q_{\mathcal{T}_i}(f_{\boldsymbol{\theta}'_i})$$

- Finn et al 2017 propose a first-order approximation, called FOMAML, which computes the gradients with respect toθ′i rather than θ

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \nabla_{\boldsymbol{\theta}'_i} \mathcal{L}^q_{\mathcal{T}_i}(f_{\boldsymbol{\theta}'_i})$$

Model Agnostic Meta learning (MAML) *(optimization based)*

- The previous form of equation work for equal number of classes so the authors adapt it to N- classes by adding output layer parameters φ_i

$$\boldsymbol{\theta}_i', \boldsymbol{\phi}_i' \leftarrow \mathrm{SGD}(\mathcal{L}^s_{\mathcal{T}_i}, \boldsymbol{\theta}, \boldsymbol{\phi}_i, \alpha, \gamma, m)$$

Also they introduce different output layer for each task and corresponding learning rate.

$$\boldsymbol{\theta} \leftarrow \mathrm{Adam}\left(\sum_i \mathcal{L}^q_{\mathcal{T}_i}(\boldsymbol{\theta}_i', \boldsymbol{\phi}_i'), \beta\right)$$

ProtoMAML (Snell et al 2017)

Prototypical networks + MAML is obtained by initializing the final layer of the classifier in each episode with these prototypical network-equivalent weights and biases ($w_c = 2\mu_c$ and $b_c = -\mu_T c \mu_c$ Triantafillou et al 2020)

Baselines

- **Majority sense Baseline** - Predicts most frequent sense in the support set
- **Nearest Neighbour classifier** - Predict sense of query as the sense of the nearest of its near neighbour from the support set - not for GLOVE
- **Episodic Fine tuning** (EF-) random initialized model with only testing time fine tuning
- **Non episodic training** (NE Baseline) - single model using merged support + query set for training , output layer is task-independent and output units is equal to number of sense classes in the dataset. During testing, it is fine tuned on support sets of each tasks (in an episodic fashion)

Experiment Setup

Dataset

SemCor corpus manually annotated corpus - 37,176 annotated sentences.

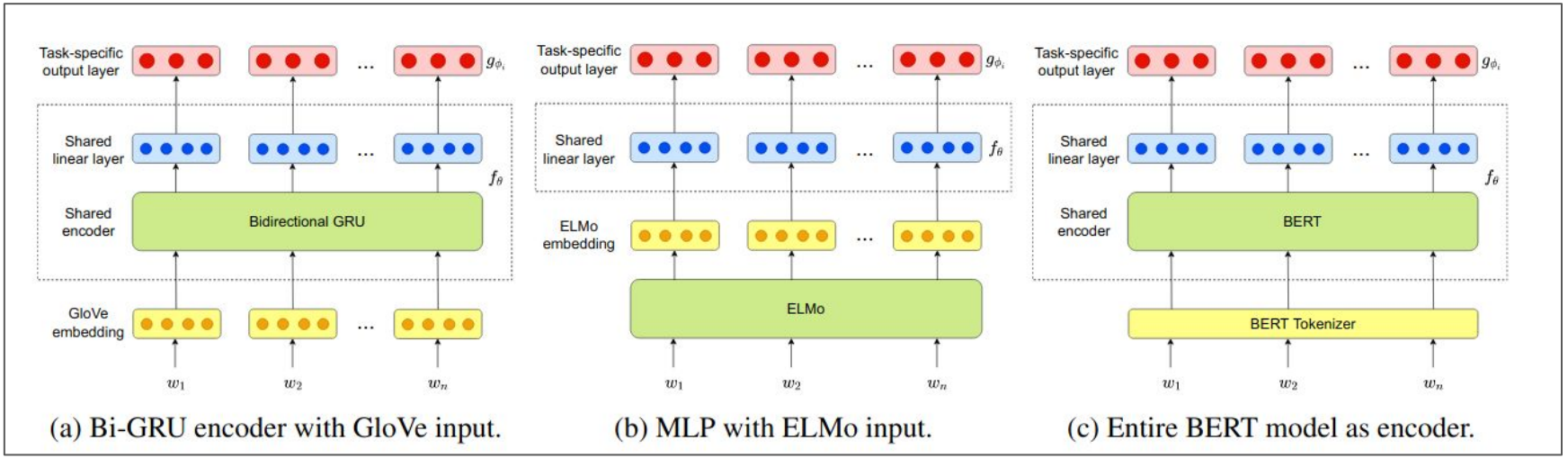
Data splits: meta-train and meta-val, meta-test 60:15:25

Support set |S| = 4,8,16,32

#distinct words in meta-train/meta-test sets :985/270, 985/259,799/197and 580/129

Experiment Setup

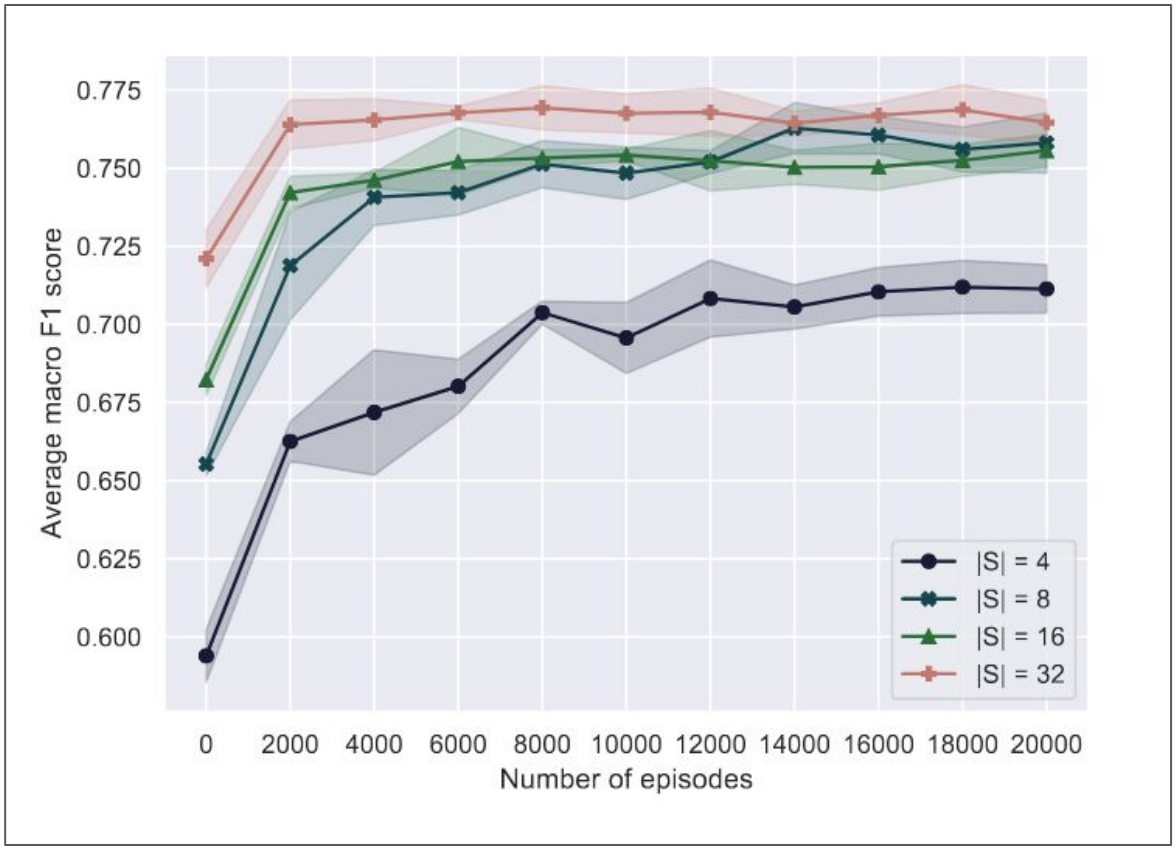
Architecture

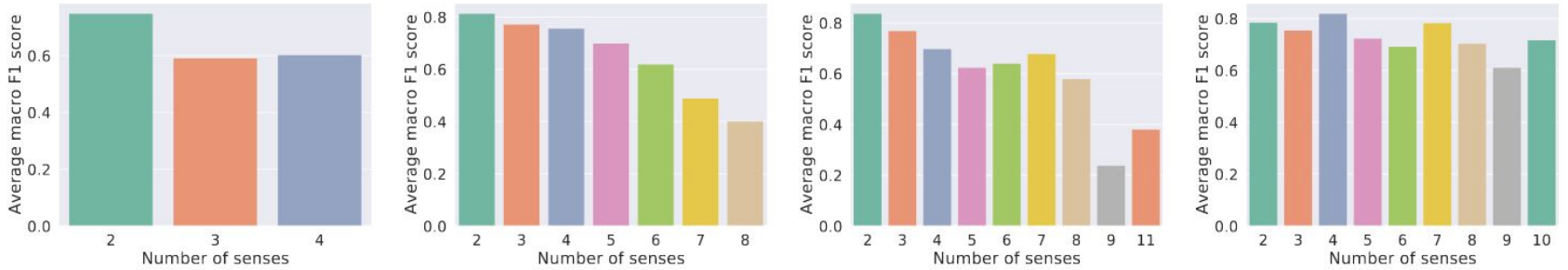


Results

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	–	–	–	–
	NE-Baseline	0.532 ± 0.007	0.507 ± 0.005	0.479 ± 0.004	0.451 ± 0.009
	EF-ProtoNet	0.522 ± 0.008	0.539 ± 0.009	0.538 ± 0.003	0.562 ± 0.005
	EF-FOMAML	0.376 ± 0.011	0.341 ± 0.002	0.321 ± 0.004	0.303 ± 0.005
	EF-ProtoFOMAML	0.519 ± 0.006	0.529 ± 0.010	0.540 ± 0.004	0.553 ± 0.009
	ProtoNet	**0.579 ± 0.004**	0.601 ± 0.003	**0.633 ± 0.008**	**0.654 ± 0.004**
	FOMAML	0.536 ± 0.007	0.418 ± 0.005	0.392 ± 0.007	0.375 ± 0.005
	ProtoFOMAML	0.577 ± 0.011	**0.616 ± 0.005**	0.626 ± 0.005	0.631 ± 0.008
ELMo+MLP	NearestNeighbor	0.624	0.641	0.645	0.654
	NE-Baseline	0.624 ± 0.013	0.640 ± 0.012	0.633 ± 0.001	0.614 ± 0.008
	EF-ProtoNet	0.609 ± 0.008	0.635 ± 0.004	0.661 ± 0.004	0.683 ± 0.003
	EF-FOMAML	0.463 ± 0.004	0.414 ± 0.006	0.383 ± 0.003	0.352 ± 0.003
	EF-ProtoFOMAML	0.604 ± 0.004	0.621 ± 0.004	0.623 ± 0.008	0.611 ± 0.005
	ProtoNet	0.656 ± 0.006	0.688 ± 0.004	0.709 ± 0.006	0.731 ± 0.006
	FOMAML	0.642 ± 0.009	0.589 ± 0.010	0.587 ± 0.012	0.575 ± 0.016
	ProtoFOMAML	**0.670 ± 0.005**	**0.700 ± 0.004**	**0.724 ± 0.003**	**0.737 ± 0.007**
BERT	NearestNeighbor	0.681	0.704	0.716	0.741
	NE-Baseline	0.467 ± 0.157	0.599 ± 0.023	0.539 ± 0.025	0.473 ± 0.015
	EF-ProtoNet	0.594 ± 0.008	0.655 ± 0.004	0.682 ± 0.005	0.721 ± 0.009
	EF-FOMAML	0.445 ± 0.009	0.522 ± 0.007	0.450 ± 0.008	0.393 ± 0.002
	EF-ProtoFOMAML	0.618 ± 0.013	0.662 ± 0.006	0.654 ± 0.009	0.665 ± 0.009
	ProtoNet	0.696 ± 0.011	0.750 ± 0.008	**0.755 ± 0.002**	**0.766 ± 0.003**
	FOMAML	0.676 ± 0.018	0.550 ± 0.011	0.476 ± 0.010	0.436 ± 0.014
	ProtoFOMAML	**0.719 ± 0.005**	**0.756 ± 0.007**	0.744 ± 0.007	0.761 ± 0.005

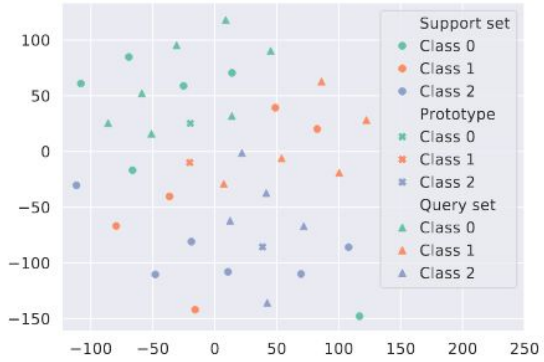
Table 1: Average macro F1 scores of the meta-test words.


0.775
0.750
0.725
0.700
0.675
0.650
0.625
0.600
Average macro F1 score
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
Number of episodes
|S| = 4
|S| = 8
|S| = 16
|S| = 32

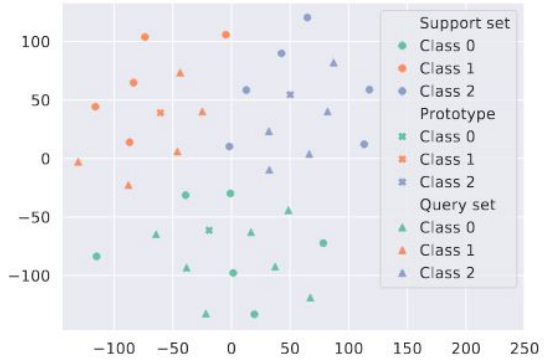


(a) $|S| = 4$ (b) $|S| = 8$ (c) $|S| = 16$ (d) $|S| = 32$

Figure 3: Bar plot of macro F1 scores averaged over words with a given number of senses.



(a)



(b)

Figure 4: t-SNE visualizations comparing ELMo embeddings (left) against representations generated by ProtoNet with ELMo+MLP (right) for the word 'field'.

Conclusion

- Meta learning appears to be helpful to handle data scarcity problem using few shot approach.
- Similarity along with optimization based meta learning better than the standalone learning paradigm.
- It could be interesting to model working for other language which have different lexical properties than English.
- Also, the case of of building such systems for multiple languages.

References

- Learning to Learn to Disambiguate:Meta-Learning for Few-Shot Word Sense Disambiguation, Holla et al 2020, Findings of the Association for Computational Linguistics: EMNLP 2020 (https://www.aclweb.org/anthology/2020.findings-emnlp.405.pdf)