Automatic Processing of User-Generated Short Texts and applications to Conversational Systems

Lara, the Meetic Chatbot

Thomas Dopierre

May 10, 2021

Laboratoire Hubert Curien
Meetic

Introduction - Meetic & Lara

Introduction – Meetic

- Dating website created in 2001
- At the origin of more than 8 millions of couples[1]
- More than 5 billion interactions between users in 2014
- Present in 15 countries, 13 languages
- First dating website to have a Chatbot

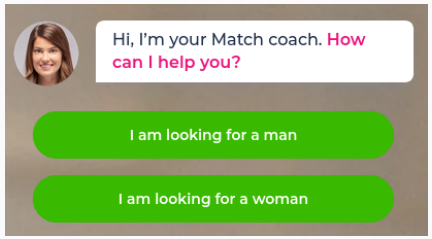
[1] https://www.meetic-group.com/who-we-are/

Lara – Goals

Born in 2017, *Lara* is a conversational agent, with multiple goals

Lara – Goals

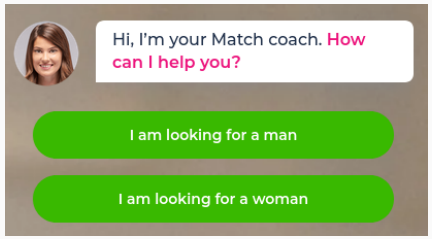
Born in 2017, *Lara* is a conversational agent, with multiple goals



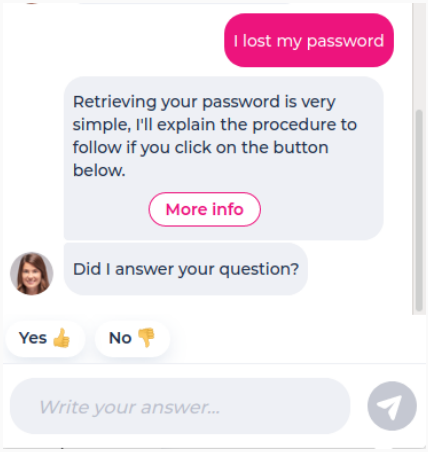
Profile Creation

Lara – Goals

Born in 2017, *Lara* is a conversational agent, with multiple goals



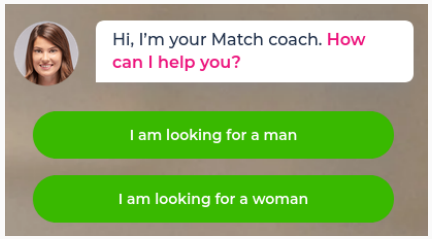
Profile Creation



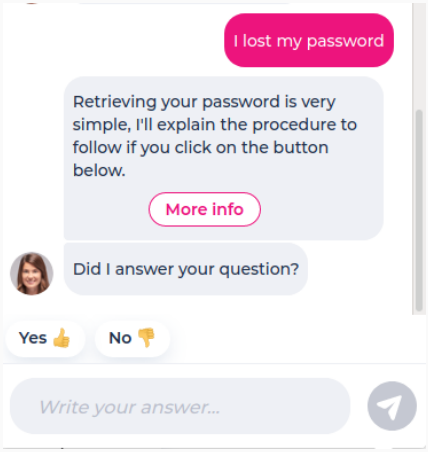
Customer Service

Lara – Goals

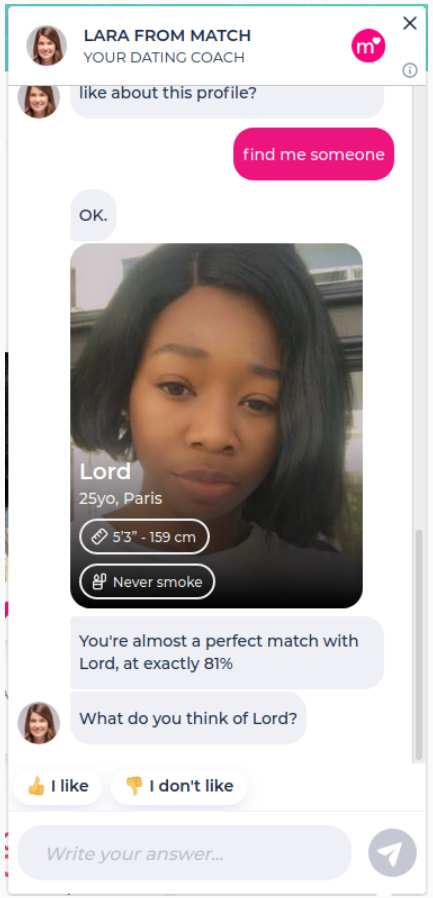
Born in 2017, *Lara* is a conversational agent, with multiple goals



Profile Creation



Customer Service

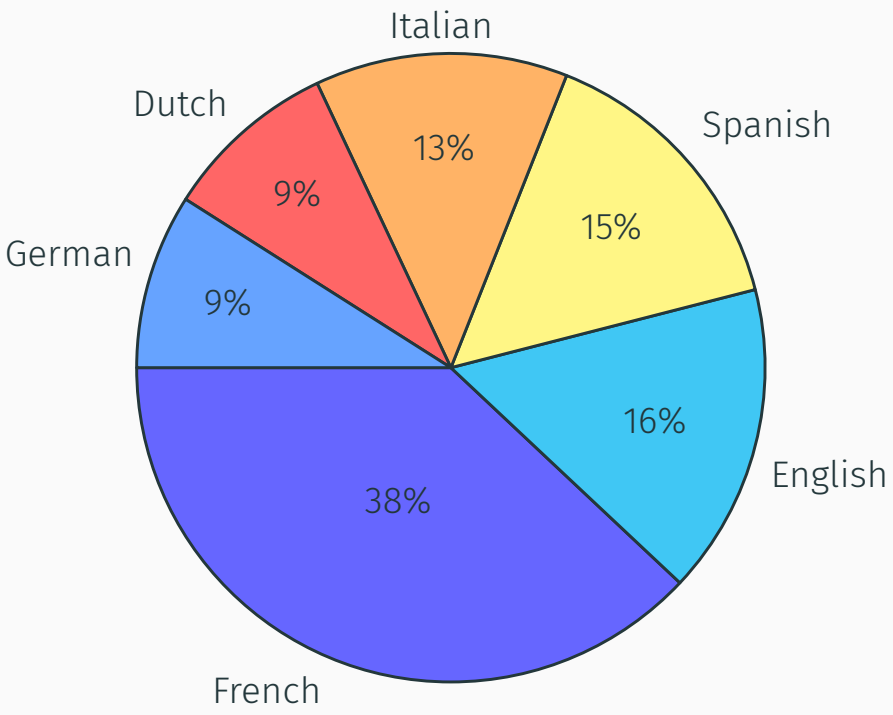


Profile Recommendation

Lara – A multilingual chatbot

Every day, Lara

- Talks to more than **50,000** users, in 6 languages (English, French, German, Spanish, Italian, Dutch)
- Presents more than **70,000** profiles



Lara – Framework

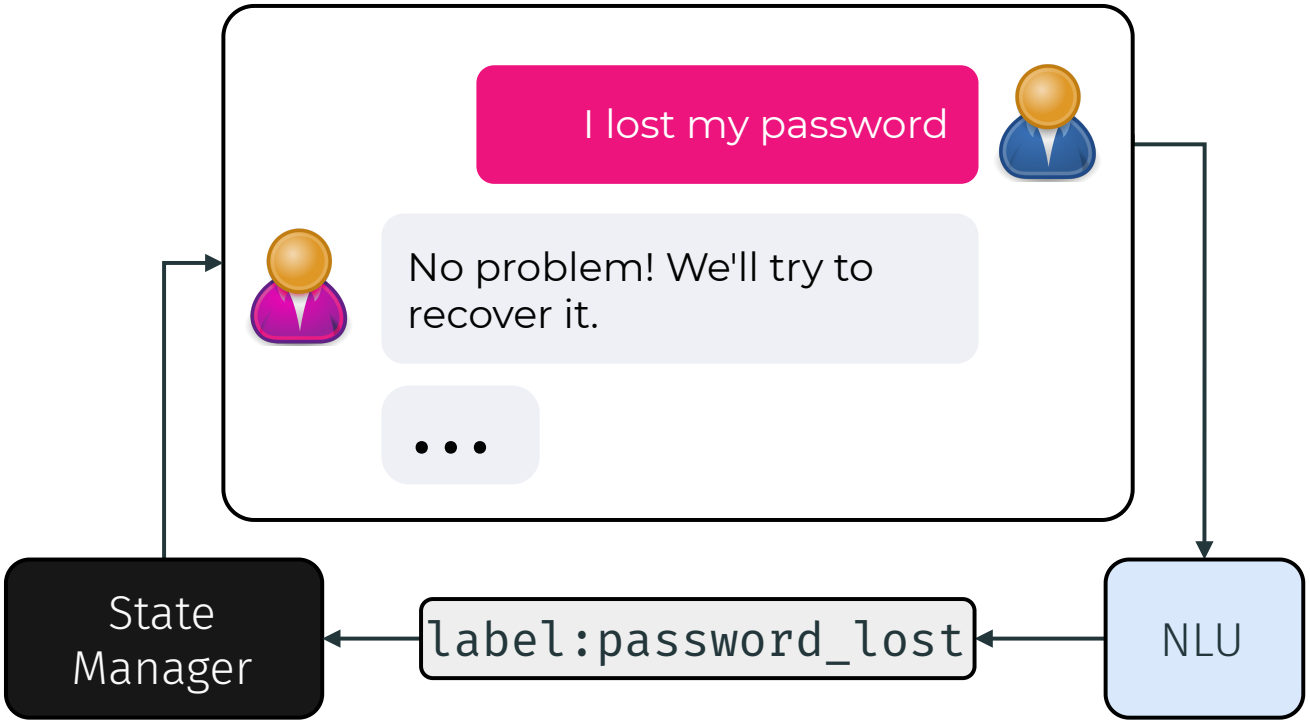


Figure 1: Simplified overview of the Lara Framework

Lara – Challenges

- Understand more than 300 different user intents
- User intents usually evolve with time
- Learn to classify using **very few samples**

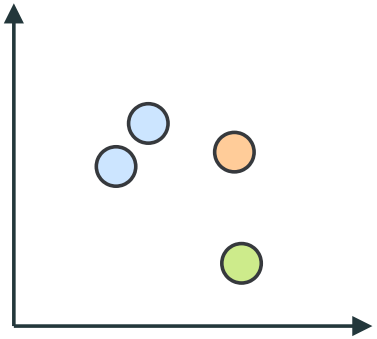
Good news: As users interact with the bot, we have some unlabeled data to work with!

Sentence Encoders

Sentence Encoding

From raw text...

...To an embedding space



Sentence encoder history

Sentence encoders have evolved in the last years

- Fixed word embeddings{GloVe [17], FastText [10], Dict2Vec [25]} + model {CNN [11], Bi-LSTM [14], GRU [30]} to embed sentences
- Contextualised embeddings with ELMo [18]
- Transformers [26]

Sentence encoder history

Sentence encoders have evolved in the last years

- Fixed word embeddings{GloVe [17], FastText [10], Dict2Vec [25]} + model {CNN [11], Bi-LSTM [14], GRU [30]} to embed sentences
- Contextualised embeddings with ELMo [18]
- Transformers [26]

Transformers have been a major break point for many tasks [3]

Choosing the right sentence encoder

Fine-tuning sentence embeddings even on small task-specific dataset can lead to large performance gains [23, 9]

Choosing the right sentence encoder

Fine-tuning sentence embeddings even on small task-specific dataset can lead to large performance gains [23, 9]

We want to use BERT, so we fine-tune it on the masked language modeling task for a few epochs on the different datasets.

Choosing the right sentence encoder

Fine-tuning sentence embeddings even on small task-specific dataset can lead to large performance gains [23, 9]

We want to use BERT, so we fine-tune it on the masked language modeling task for a few epochs on the different datasets.

MLM is fully unsupervised, so we can use the whole datasets!

Choosing the right sentence encoder

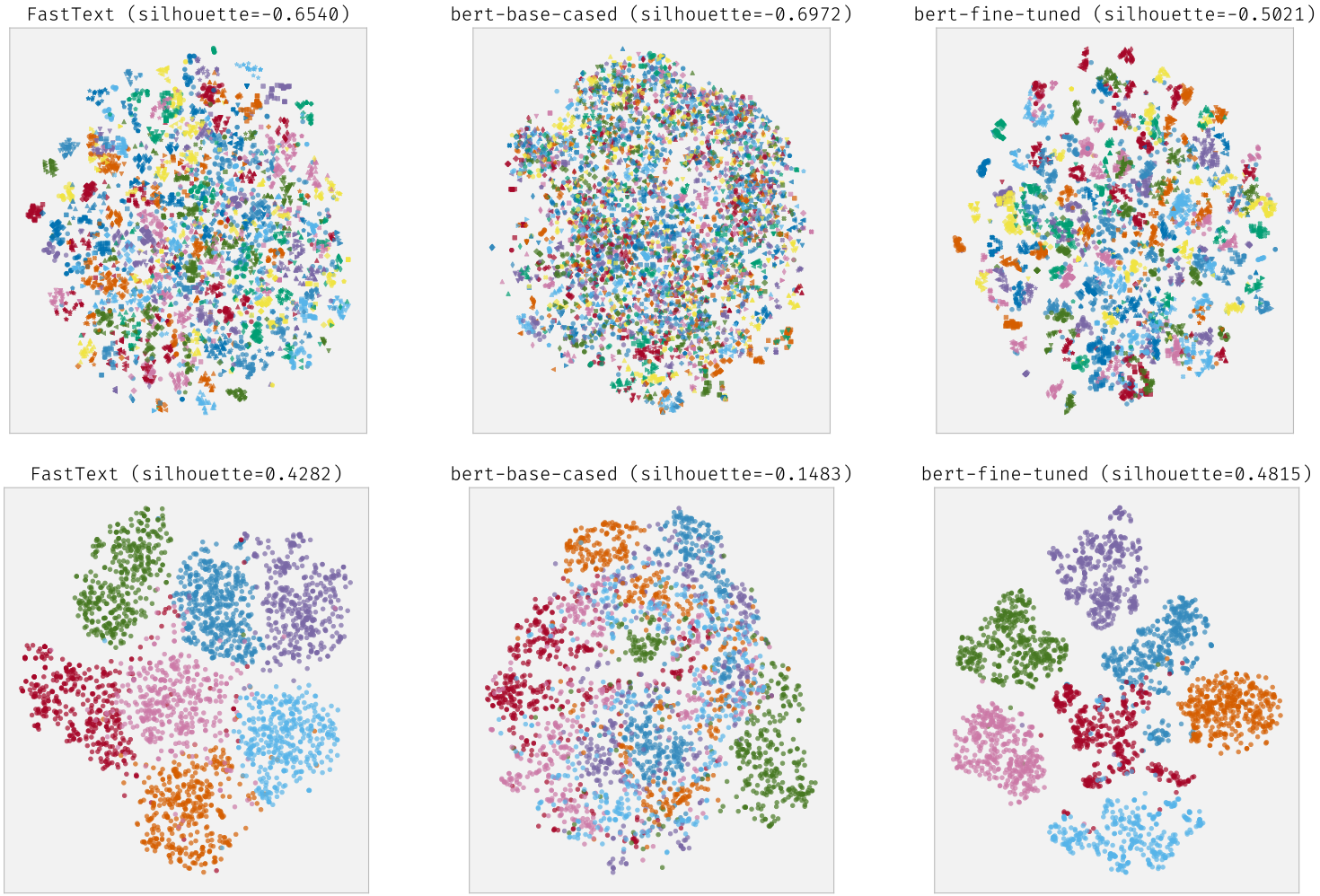


Figure 2: T-SNE representations of the Snips [2] and OOS [12] datasets using different encoders

Few-Shot Learning vs. Pseudo-Labeling

A problem overview

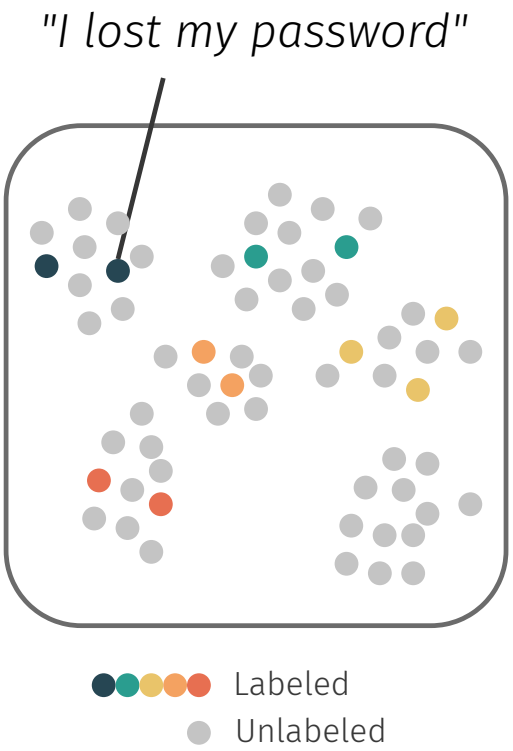


Figure 3: Data overview

We have both labeled and unlabeled data at hand

$|\text{LABELED}| \ll |\text{UNLABELED}|$

We want to train a classifier using labeled (and maybe unlabeled) data

Pseudo-Labeling

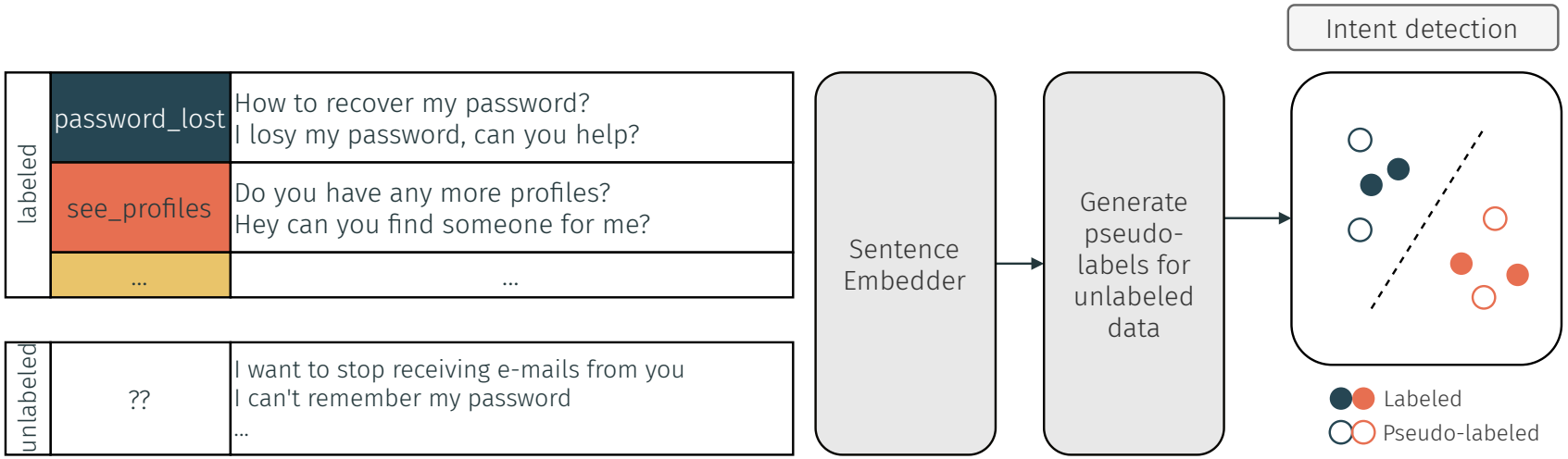


Figure 4: Pseudo-Labeling Framework

Goal: Assign pseudo-labels to unlabeled data, train a model using both labels and pseudo-labels

Evaluation Metric: correctness of pseudo-labels & Intent Detection performance

Few-Shot Learning as a Meta-Learning process

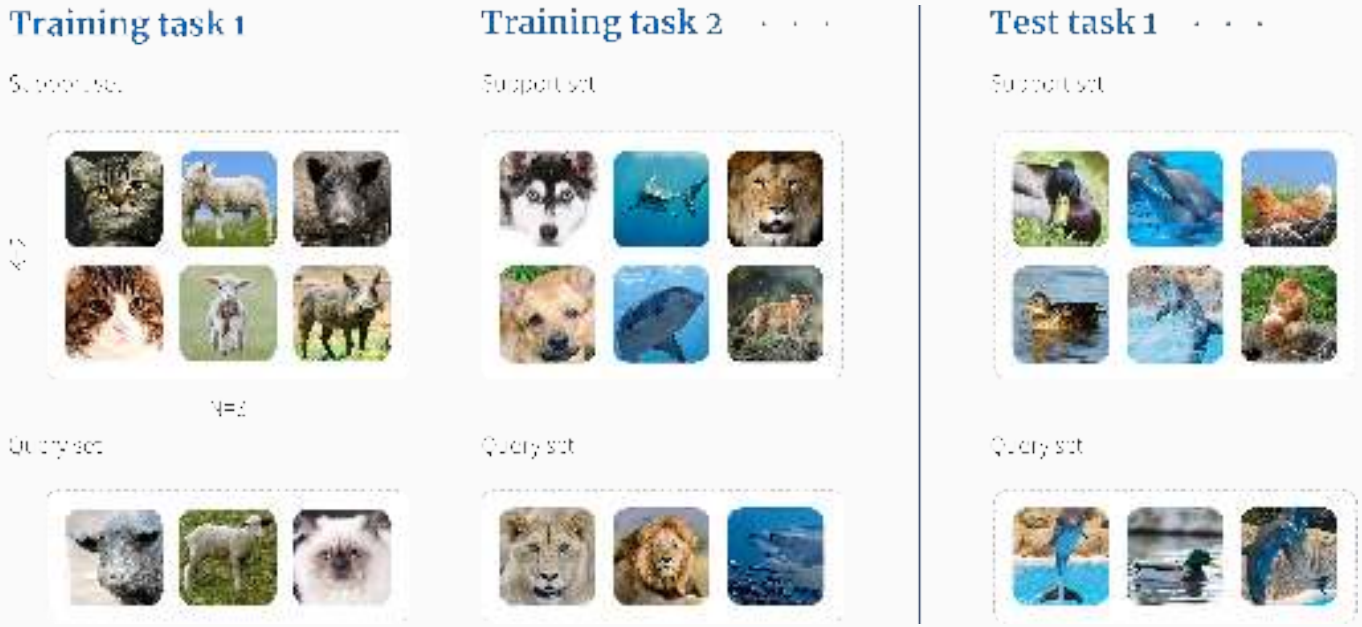


Figure 5: Episode-like process in Few-Shot Image Classification

- Episode-like process [22, 28], where support points are used to predict query points
- Train, valid and test labels are different

Goal: Train a end-to-end model, which will be able to generalise to test classes using only few shots

Few-Shot Learning with Prototypical Networks

Idea: Represent each class as the average vector (prototype) of its support points.

Prediction: The label assigned to a query point x is the one for which the euclidean distance to the prototypes is the lowest.

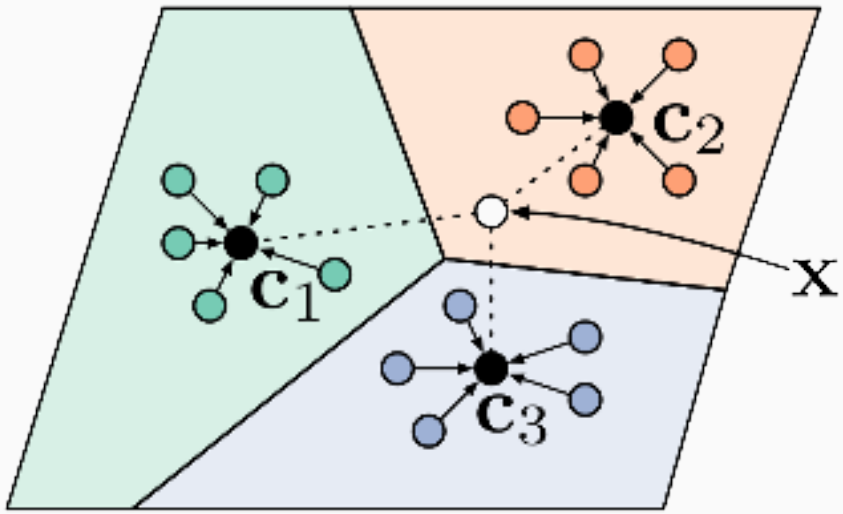
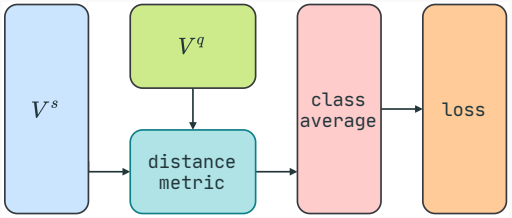


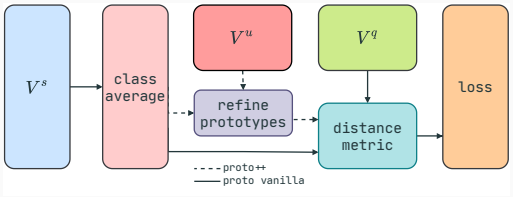
Figure 6: Prototypical Networks

Prototypical Networks for Few-shot Learning [22]

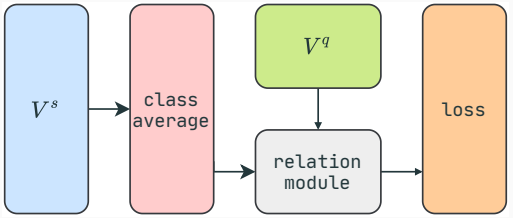
Other Few-Shot Learning Methods



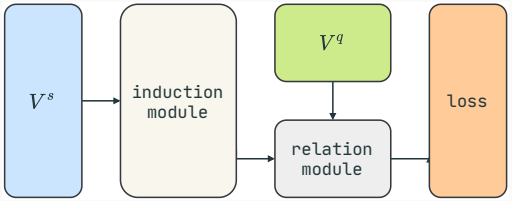
Matching Network [28]



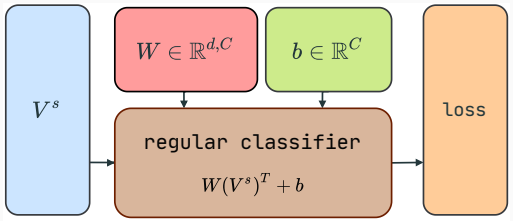
Prototypical Network [22]



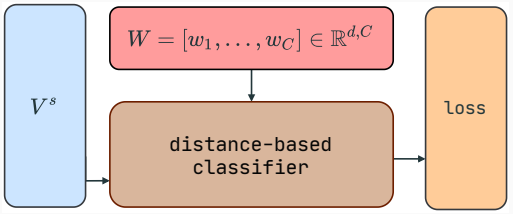
Relation Network [24]



Induction Network [8]



Baseline Network



Baseline++ Network [1]

Figure 7: Few-Shot Text Classification methods used in our experiments

In this presentation

We have contributions on both approaches

In this presentation

We have contributions on both approaches

- A novel method for pseudo-labeling using hierarchical clustering

In this presentation

We have contributions on both approaches

- A novel method for pseudo-labeling using hierarchical clustering
- An updated benchmark of end-to-end few-shot text classification methods using a common text encoder

In this presentation

We have contributions on both approaches

- A novel method for pseudo-labeling using hierarchical clustering
- An updated benchmark of end-to-end few-shot text classification methods using a common text encoder
- A novel extension of Prototypical Networks using unlabeled data and diverse paraphrasing

Pseudo-Labeling with Hierarchical Clustering

Pseudo-Labeling with Hierarchical Clustering

Idea: Use hierarchical clustering to produce pseudo-labels for unlabeled data points

Goals: Avoid using too much hyper-parameters. Avoid assigning pseudo-labels to all data points, (some unlabeled data points might be noise)

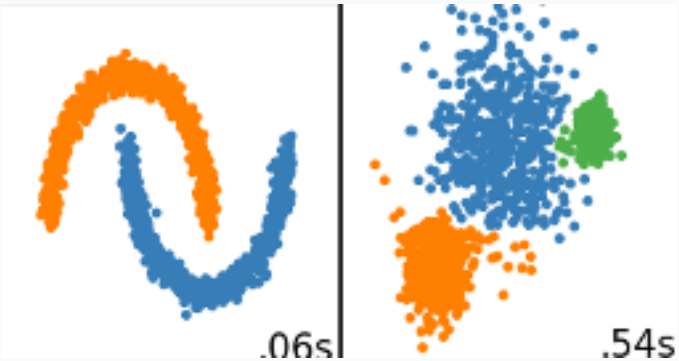


Figure 8: Examples of clusters obtained by hierarchical clustering[2]

[1] https://scikit-learn.org/stable/modules/clustering.html

Pseudo-Labeling with Hierarchical Clustering – Step (a)

Input: Labeled and unlabeled data points

Process: Iteratively constructs a tree using hierarchical clustering. Binds the two closest clusters (according to Ward's method) at each iteration

Output: A tree structure, where each non-leaf node has exactly two children, representing a merge between two nodes at some iteration.

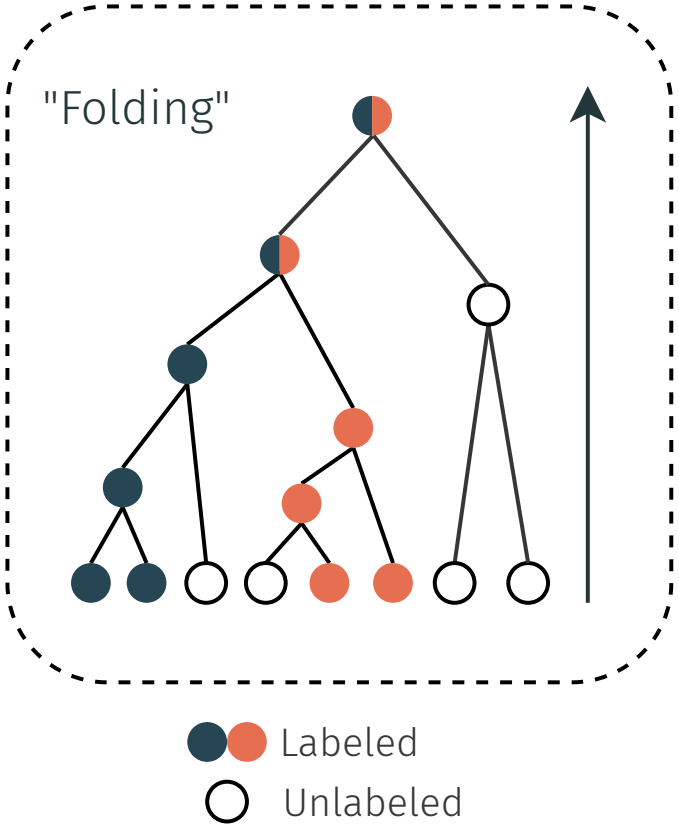


Figure 9: Two-fold Pseudo-Labeling: Step (a)

Pseudo-Labeling with Hierarchical Clustering – Step (b)

Process: Iteratively de-constructs the tree obtained at step (a), splitting clusters into their two sub-clusters until (1) all generated clusters contain either no labeled data points, or (2) some labeled data points with a unique label.

Case (1) Discard data points.

Case (2) Assign the unique label to all unlabeled data points in the same cluster.

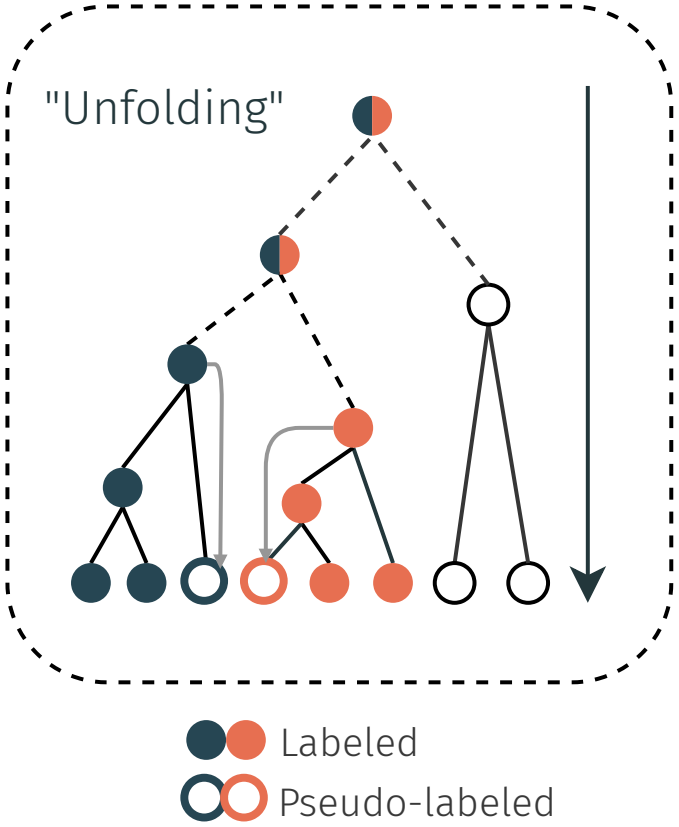


Figure 10: Two-fold Pseudo-Labeling: Step (b)

Datasets

	#sentences	#classes (train/valid/test)	#sentences/class	#tokens/sentence
Snips [2]	$14,484$	$2/2/3$	$2,069 \pm 21$	9.0 ± 3.2
OOS [12]	$23,700$	$53/52/46$	157 ± 85	8.5 ± 3.3
Liu [15]	$25,478$	$19/18/17$	472 ± 823	7.5 ± 3.4
R8 [19]	$7,685$	$3/2/3$	961 ± 1303	102 ± 117

Table 1: Datasets

Results

	Snips		OOS		Liu		R8	
Model \ Shots	1	5	1	5	1	5	1	5
Proto	0.913	0.947	0.648	0.829	0.515	0.694	0.598	0.725
Proto++	0.863	0.920	0.416	0.640	0.472	0.605	0.566	0.628
nKNN	0.945	0.985	0.725	0.902	0.505	0.645	0.758	0.885
Fold/Unfold	**0.988***	**0.992**	0.878	**0.945***	0.537	**0.728**	**0.817**	0.901
+ weights	0.987	**0.992**	**0.880***	0.944	**0.539**	0.726	**0.817**	**0.902***
Aggregated	0.983	0.992	0.839	0.927	0.585	0.747	0.813	0.894
Shots-Only	0.948	0.987	0.714	0.901	0.443	0.756	0.738	0.892

Table 2: F1-score of Intent Detection system fed with pseudo-labels from various methods.

According to experiments, our method:

- is the best on all datasets
- is robust to the large number of classes (OOS) and the more general case of text classification (R8)

This work was published as long paper at COLING2020: **Few-shot Pseudo-Labeling for Intent Detection [6]**

End-To-End Few-Shot Learning Methods

End-To-End Few-Shot Learning Methods

Motivation: Find an End-To-End Few-Shot Learning method which takes into account unlabeled data

First: We need to study state-of-the-art Few-Shot Text Classification methods!

Problem: Methods in the literature do not use the same sentence encoder...

Model	Mean Acc
Matching Networks [28]	65.73
Prototypical Networks [22]	68.17
Graph Network [20]	82.61
Relation Network [24]	83.07
SNAIL [16]	82.57
ROBUSTTC-FSL [29]	83.12
Induction Networks [8]	85.63

Table 3: Reported Results of Few-Shot Text Classification models on the ARSC dataset

Idea: Compare methods using the same transformer-based encoder

Few-Shot Learning: A reality check on ARSC

Model	Configuration		Mean binary accuracy		
	Metric	Relation module	Original encoder †	BERT as encoder (↗ or ↘ w.r.t. original encoder)	BERT + Sample shots
Matching Network (Vinyals et al., 2016)	euclid.	N/A	–	81.2	82.9
	cosine	N/A	65.7	81.9 (↗)	83.3
Prototypical Network (Snell et al., 2017)	euclid.	N/A	68.2	80.0 (↗)	82.6
	cosine	N/A	–	81.7	83.5
Proto++ (Ren et al., 2018)	euclid.	N/A	📷	82.4	**84.0**
	cosine	N/A	📷	**82.6**	83.6
Relation Network (Sung et al., 2018)	N/A	base	–	81.0	82.9
	N/A	ntl	83.1	81.7 (↘)	83.3
Induction Network (Geng et al., 2019)	N/A	ntl	85.6	79.3 (↘)	80.3
Baseline	N/A	N/A	📷	80.7	79.8
Baseline++	euclid.	N/A	–	81.9	82.2
	cosine	N/A	📷	79.7	81.1

Figure 11: Reality-check of Few-Shot Learning systems on the ARSC dataset

Few-Shot Learning: A reality check for Intent Detection

	Metric	Relation Module	Liu				OOS				TREC28			
			2	3	4	5	2	3	4	5	2	3	4	5
Matching Net (2016)	cosine	-	93.3	87.9	84.8	81.0	96.8	95.8	95.1	94.7	81.6	75.4	68.5	63.5
Proto Net (2017)	euclid.	-	97.4	95.3	93.4	91.8	**99.5**	99.0	98.7	98.4	**92.6**	**87.6**	82.0	**79.2**
Proto++ (2018)	euclid.	-	**97.7**	**95.7**	**93.7**	**92.2**	99.5	**99.1**	**98.8**	**98.5**	91.7	84.9	**82.0**	76.8
Relation Net (2018)	-	ntl	87.4	80.1	74.3	69.0	90.9	84.2	82.0	77.8	74.7	62.5	57.7	48.6
Induction Net (2019)	-	ntl	73.9	57.9	52.6	40.6	74.9	59.3	50.9	43.8	70.3	49.6	41.9	33.9
Baseline	-	-	94.3	89.0	84.1	79.8	99.1	98.5	97.7	97.2	90.5	83.6	79.3	75.7
Baseline++	euclid.	-	93.1	87.6	81.4	78.1	95.8	93.3	92.1	90.6	87.7	78.3	72.5	69.1
	cosine	-	93.1	86.8	81.0	75.1	98.9	97.9	96.8	96.1	86.7	78.2	72.1	70.0

Table 4: Few-Shot Learning methods, tested on 3 intent detection datasets, provided the same transformer encoder.

Main findings:

- Prototypical Networks reclaim the state-of-the-art when equipped with a transformer
- The distance metric plays an important role
- A traditional classifier yields competitive results

This work was published as long paper EACL2021: **A Neural Few-Shot Text Classification Reality Check [4]**

PROTAUGMENT: Intent Detection Meta-Learning through Unsupervised Diverse Paraphrasing

Motivations

- Extend prototypical networks