SPOKEN LANGUAGE UNDERSTANDING FOR INDIC

Anurag Dwarakanath, Applied Science Manager,

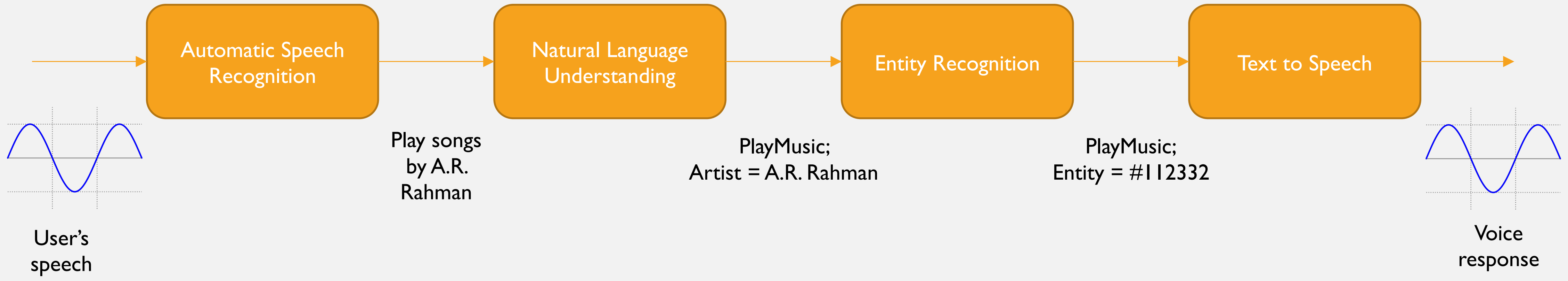
Alexa AI

AGENDA

- An overview of SLU systems
  - Current State of the Art architecture;
- The Future
  - Challenge 1: Tackling Indic data in transliterated form
    - Multi-lingual representation of transliterated data
  - Challenge 2: Supporting new Languages
    - Continual Language Learning
  - Challenge 3: Building Robust Machine Learning
    - Robustness to labels and variations in input

AN OVERVIEW OF SPOKEN LANGUAGE UNDERSTANDING SYSTEMS

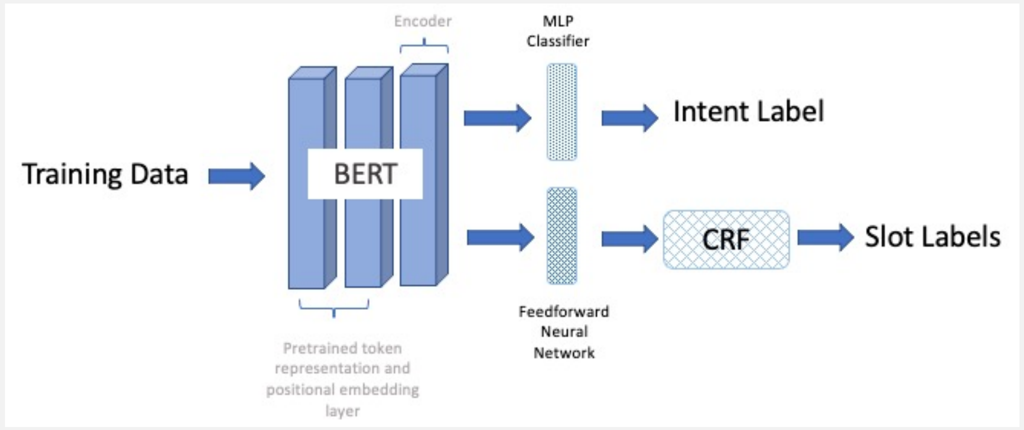
- Typical Spoken Language Understanding (SLU) systems follow the 'pipeline' architecture to address users' queries.


Automatic Speech Recognition
Natural Language Understanding
Entity Recognition
Text to Speech
User's speech
Play songs by A.R. Rahman
PlayMusic; Artist = A.R. Rahman
PlayMusic; Entity = #112332
Voice response

Pipeline Architecture. Source: Alan Packer, "Natural Language Understanding in Alexa", 2019. https://www.youtube.com/watch?v=U1yT_4xcglY
Image source: https://commons.wikimedia.org/wiki/File:Simple_sine_wave.svg (license: CC-BY-SA)

AN OVERVIEW OF SPOKEN LANGUAGE UNDERSTANDING ARCHITECTURE

- Focusing on Natural Language Understanding, our mental model for the ML task is a joint Intent Classification and Slot-Filling [Chen, Qian, Zhu Zhuo, and Wen Wang 2019]


Encoder
MLP Classifier
Intent Label
Training Data
BERT
CRF
Slot Labels
Feedforward Neural Network
Pretrained token representation and positional embedding layer

$$y^i = softmax(W^i h_1 + b)$$

Where y^i is the Intent label, h_1 is the hidden representation of the CLS token

$$y_n^s = softmax(W^s h_n + b), n \in 1 \ldots N$$

Where y_n^s is the Slot label for n^{th} sub-word, h_n is the hidden representation of the n^{th} sub-word token

$$p(y^i, y^s|x) = p(y^i|x) \prod_{n=1}^{N} p(y_n^s|x)$$

Optimise joint probability of Intent & Slot labels. Notice this is a straight usage of Markov assumption

Chen, Qian, Zhu Zhuo, and Wen Wang. "Bert for joint intent classification and slot filling." *arXiv preprint arXiv:1902.10909* (2019).

CHALLENGE 1: INDIC LANGUAGE REPRESENTATION

- It’s well known that users in the Indic region make ample use of Code-Mixed utterances:
  - A.R. Rahman के songs बजाओ – translates to ‘play songs of A.R. Rahman’
- However, different Indic languages are used to interact – much beyond Hindi – including Punjabi, Tamil, Telugu, Marathi, etc.
- A key insight is that Indic language usage is more likely in Entity Names
  - E.g. – ‘play the movie ಇಕ್ಕಟ್’ - translates to ‘play the movie Ikkat’ (a recent Kannada movie).
  - E.g. – ‘show me latest చీర designs’ – translates to ‘show me latest designs of Sari’ (చీర is Sari in Telugu).

CHALLENGE 1: INDIC LANGUAGE REPRESENTATION

- When user says ‘play the movie ಇಕ್ಕಟ್’ to an English ASR model
  - The input to SLU comes as - ‘play the movie ikkat’
  - Notice the transliterated text ‘ikkat’
- Such inputs are common in non-Alexa settings as well – search queries, social media, chats
- Question: How do we build machine learning classifiers that can appropriately handle transliterated text?

CHALLENGE 1: DO MULTI-LINGUAL MODELS HELP?

- A natural direction would be to use multi-lingual pre-trained models, such as mBERT, XLM-R?
  - These models have been trained on multi-lingual text and build semantically similar representations of concepts between languages [Pires 2019].
  - E.g. these models have similar representations for 'बजाओ' & 'play'.
- For the figure on the right, we plot the top three dimensions of sentence embeddings through SVD.
  - Red: Sentences only in Devanagiri
  - Green: Sentences in only Latin
  - Blue: Sentences in Devangiri & Latin

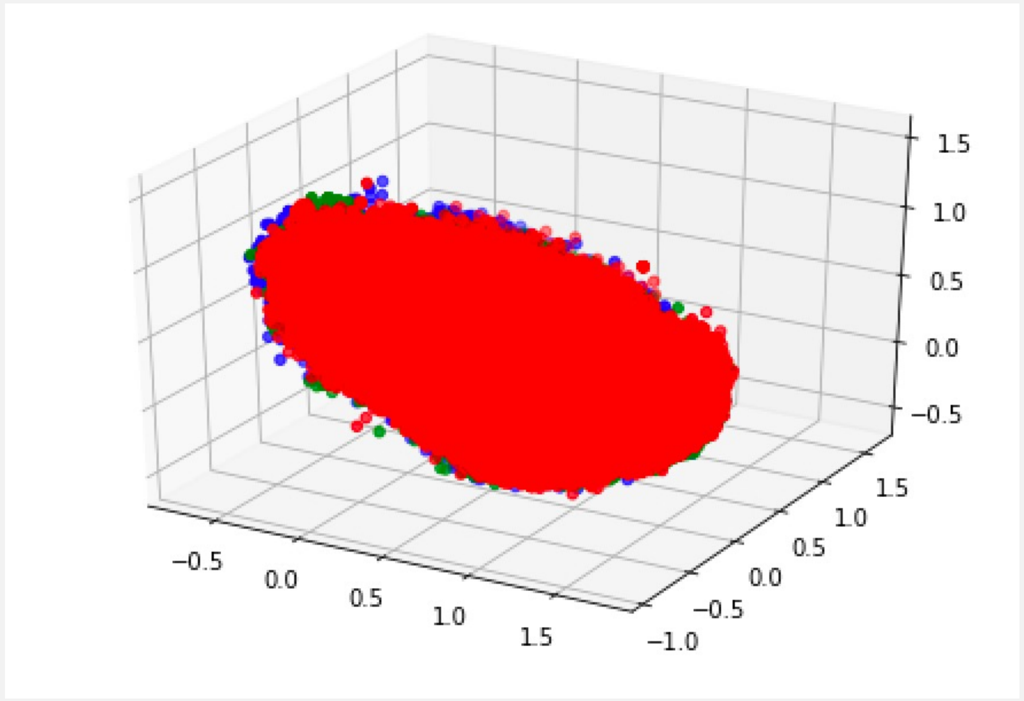


Figure: 3D plot of sentence embeddings from a multi-lingual model. Notice large overlap between sentences in code-mixed form

Pires, Telmo, Eva Schlinger, and Dan Garrette. "How multilingual is multilingual BERT?." *arXiv preprint arXiv:1906.01502* (2019).

CHALLENGE 1: DO MULTI-LINGUAL MODELS HELP?

- An experiment on three testsets spanning transliterated & monolingual utterances was used to test for classification errors (shown in adjoining table)
  - -ve implies errors were reduced.
- Surprisingly, these multi-lingual models do not handle transliterated text well.
  - We see that creating training data with explicit transliteration helped.
  - Usage of a multi-lingual model deteriorated performance.
- Insight: multi-lingual models see transliterated text as a whole new language!
  - Also cited in [Pires 2019]

TestSet	XLM-R (multi-lingual model)	Monolingual model with Hindi training data transliterated into English	XLM-R with with Hindi training data transliterated into English
Hindi Transliterated into English	+4.76%	-15.29%	-15.72%
Monolingual – English	+1.39%	+0.93%	+3.03%
Monolingual - Hindi	-28.89%	-2.52%	-30.88%

Pires, Telmo, Eva Schlinger, and Dan Garrette. "How multilingual is multilingual BERT?." *arXiv preprint arXiv:1906.01502* (2019).

DOING BETTER THAN TRANSLITERATION BASED TRAINING

- While results are better by using specific training data with transliterated text
  - Not an elegant solution
  - Requires creation of new training data as new languages are supported
  - Needs the dynamic expansion of the Vocabulary
    - E.g. to support 'cheera'
- Few Directions to pursue
  - Incorporation of phonetic features [Dalmia et. al 2019]
  - Character level tokenisers [Clark et. al 2021]

Dalmia, S., Li, X., Black, A. W., & Metze, F. (2019). *PHONEME LEVEL LANGUAGE MODELS FOR SEQUENCE BASED LOWRESOURCE ASR.*
Clark, Jonathan H., et al. "Canine: Pre-training an efficient tokenization-free encoder for language representation." *arXiv preprint* (2021).

CHALLENGE 2: CONTINUAL LANGUAGE LEARNING

- Multi-lingual models (such as mBERT or XLM-R) have been shown to improve performance in zero-shot cross lingual transfer [Shijie, 2019; Conneau, 2019]
  - These models also improve performance on resource rich language through transfer learning [Pires et al, 2019]
- Challenge: How do we increase the language support for new Languages?
- E.g. mBERT is trained on 104 languages, but not on many Indic Languages.
  - If we want to support new languages, such as Kannada, how can we build the support?
  - E.g. input is 'A.R. Rahman ಹಾಡುಗಳನ್ನು ಪ್ಲೇ ಮಾಡಿ'
  - Current practice is to build a pre-trained model with support for such languages
    - Extremely wasteful!

Wu, Shijie, and Mark Dredze. "Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT." *arXiv preprint arXiv:1904.09077* (2019).
Conneau, Alexis, et al. "Unsupervised cross-lingual representation learning at scale." *arXiv preprint arXiv:1911.02116* (2019).
Pires, Telmo, Eva Schlinger, and Dan Garrette. "How multilingual is multilingual BERT?." *arXiv preprint arXiv:1906.01502* (2019).

CHALLENGE 2: CONTINUAL LANGUAGE LEARNING

- Can we take a pre-trained model which supports X languages and extend it to Y new languages?
- Analogous to how humans can learn new languages without needing to re-learn already known languages.
- We model this problem as 'Continual Language Learning'. 2 key problems to solve:
  - Extending Vocabulary support
    - The script of the new language script could be completely new
  - Catastrophic forgetting
    - Ensuring what has been learnt in the pre-trained model is not forgotten

CONTINUAL LANGUAGE LEARNING: VOCABULARY EXTENSION

- Naïve Approach:
  - Include new tokens from the new languages and train standard Masked Language Model training objective
- Vocabulary Substitution (Garcia et al 2021)
  - Build new vocabulary with N+1 languages. Retain embeddings of older tokens.
  - Requires access to original data used
- Vocabulary Extension (Pfeiffer, Jonas, et al. 2020)
  - Create tokens for new language. Re-use lexically overlapping embeddings.
- Tokenisation Free Methods (Xue, Linting, et al, 2021)
  - Byte level techniques that remove dependency of separate tokenisation

Garcia, Xavier, et al. "Towards Continual Learning for Multilingual Machine Translation via Vocabulary Substitution." *arXiv preprint arXiv:2103.06799* (2021).
Pfeiffer, Jonas, et al. "Unks everywhere: Adapting multilingual language models to new scripts." arXiv preprint arXiv:2012.15562 (2020).
Xue, Linting, et al. "ByT5: Towards a token-free future with pre-trained byte-to-byte models." arXiv preprint arXiv:2105.13626 (2021).

CONTINUAL LANGUAGE LEARNING: CATASTROPHIC FORGETTING

- Naïve approach:
  - Continue the standard Masked Language Modeling (MLM) objective with the new data.
  - This may work since the pre-trained model inherently act as a regulariser [McRae et al 1993]
- Knowledge Distillation [Castellucci et al 2021]
  - Use a the model trained on older languages as a teacher model and train the new model over new languages as a student model

Knowledge Distillation formulation

$$\mathcal{L}_{KD}(x) = -\sum_i y_i^t(x) \log y_i^s(x)$$

Where $\mathcal{L}_{KD}(x)$ is the loss; y^t is the label from the teacher and y^s is the label from the student

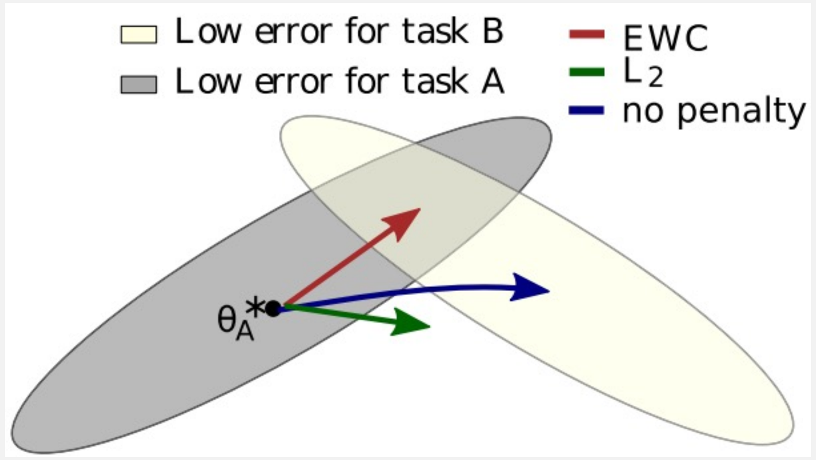
$$y_i(x) = \frac{exp(d_i/T)}{\sum_j exp(d_j/T)}$$

Unlike the standard cross-entropy loss where $y_i \in \{1 \ldots J\}$, in knowledge distillation, y_i is a probability score

Ken McRae and Phil A Hetherington, Catastrophic Interference is Eliminated in Pre-Trained Networks , 1993,
Castellucci, Giuseppe, et al. "Learning to Solve NLP Tasks in an Incremental Number of Languages.", ACL-IJCNLP 2021

CONTINUAL LANGUAGE LEARNING: CATASTROPHIC FORGETTING

- Elastic Weight Consolidation [Kirkpatrick et al 2017]
  - The method identifies those weights that are important for the old tasks and has a high regularization on them
  - This prevents the network from making drastic changes to the important weights
  - As the neural network is over-parameterized, it is likely that parameters optimized for the new task is close to that optimized for the older tasks.



$$\log p(\theta|\mathcal{D}) = \log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A) - \log p(\mathcal{D}_B)$$

Where Θ are the model parameters; D_B is the data for the new task; D_A is the data for the older task

James Kirkpatrick et al, "Overcoming catastrophic forgetting in neural networks", PNAS, 2017

CHALLENGE 3: ROBUST MACHINE LEARNING

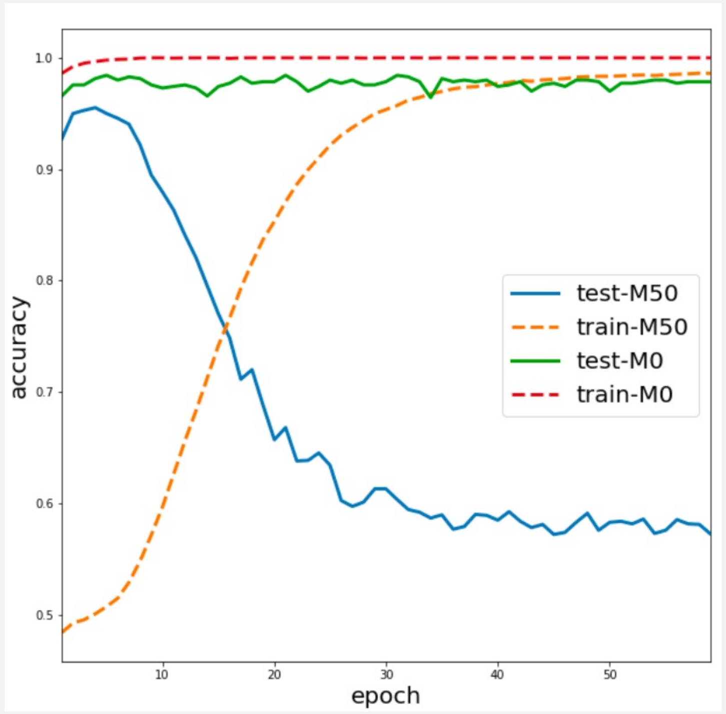
- Noise in Data is abundant.
  - Noise both in supervised training data (label noise) & un-supervised
- Label noise can occur for numerous reasons:
  - Manual Errors in Annotation:
    - “what’s the latest movie” -> gets annotated as ‘play movie’ intent.
  - Genuine Ambiguities:
    - ‘play Rang de Basanti’ -> is it a Movie or a Song?
  - Ambiguous requests:
    - “hum aapke hain kaun” -> should we play the movie or answer the question?
  - Semi-supervised techniques that generate data & labels synthetically
    - There is inherent noise/inaccuracies in the method employed

CHALLENGE 3: ROBUST MACHINE LEARNING

- Noise in user utterances:
  - Spoken Language variations:
    - “I want to learn English no Spanish” -> Users cannot backspace in voice ☺
  - Incorrect Grammar
    - “please Amit call”
  - Background noise
    - “could you [background words] play …”

ROBUST MACHINE LEARNING: IMPACT OF NOISE

- In Open Source datasets, noise in data has been found to be between 8% to 30% [Song et al 2020]
- Current Deep Neural Networks are over-parameterized
  - Number of model parameters exceed the number of data-points
- Noise in data leads to memorization [Zhang et al 2016].
  - Ending up with poor generalisation



Train and test curves on SNIPs dataset with 0% noise (M0) and 50% noise (M50)

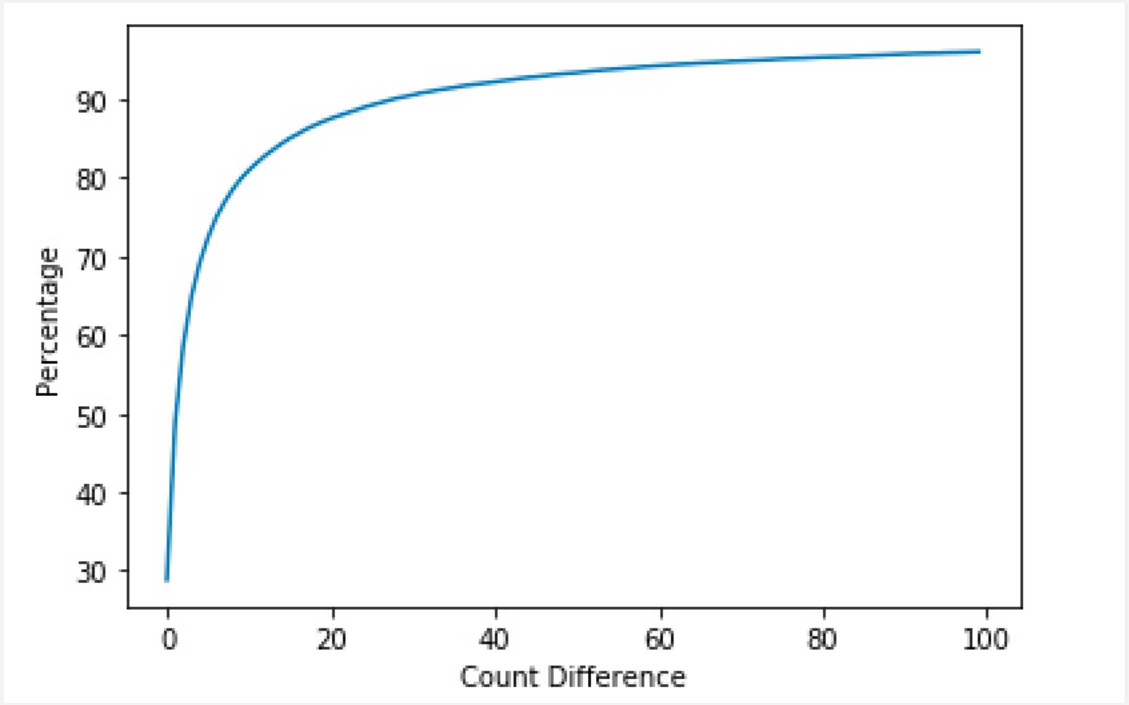
Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understand- ing deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.
Hwanjun Song, Minseok Kim, Dongmin Park, Jae-Gil Lee, "Learning from Noisy Labels with Deep Neural Networks: A Survey", Oct 2020, arXiv:2007.08199

NOISE MITIGATION TECHNIQUES

- Broadly 2 types of Noise Mitigation Techniques:
  - Correct the data.
    - Effort intensive if to be done manually
    - Can employ heuristic techniques for automated correction
  - Employ learning mechanisms that are robust to noise
    - The algorithm will employ methods to determine whether to train on a given data point

NOISE MITIGATION TECHNIQUES: CORRECT DATA

- Class 1: Correct the data.
  - Majority Voting as a simple technique
    - Choose the label where the majority votes reside.
    - However, in most cases, it is not easy to clearly segregate.
    - In the plot on the right, we see that 70% of the noisy data has less than 3 number of differences
    - Implies will have high entropy.



Cumulative plot of % of training data with number of conflicting labels on a real-life dataset

NOISE MITIGATION TECHNIQUES: LEARNING ALGORITHMS

- Noise Layer [Jindal et al, 2019]
  - A non-linear layer (called the noise layer) is added to the base neural network model.
  - The base layer and the noise layer is jointly trained. With appropriate regularization, the noise layer overfits on the label noise.
  - At inference time, the noise layer is removed.
- Robust Loss [Ma et al, 2020]
  - Uses Normalised Cross Entropy as a loss function that is robust to mild levels of label noise (majority should still be correct class).
  - To prevent underfitting, introduces the Active – Passive Loss formulation
    - Active Loss function is one that maximises the probability for the correct class. E.g. Cross-entropy
    - Passive Loss function is one that minimises the probability for all other classes. E.g. Reverse Cross Entropy.

Noise Layer Formulation

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n} \log\left[\sigma\left(\Psi \times p\left(y|\mathbf{X}_i;\Theta\right)\right)\right]_{y_i} + \frac{1}{2}\lambda||\Psi||_2^2.$$

Where ψ is the noise layer. $p(y|\boldsymbol{X}_i;\ \theta)$ is the base model

Ishan Jindal, Daniel Pressel, Brian Lester, and Matthew Nokleby. An effective label noise model for DNN text classification NAACL, 2019.
Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. ICLR 2020

NOISE MITIGATION TECHNIQUES: LEARNING ALGORITHMS

- LIMIT [Harutyunyan, et al, 2020]
  - Adds Information theoretic regularization term
  - Shows that reducing the mutual information between weights and labels is akin to making the algorithm robust.
- Early Stopping [Li et al, 2020]
  - Use error calculated on validation set as a proxy for generalization
  - Note that the validation set is also noisy.

Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. 2020. Improving generalization by controlling label-noise information in neural network weights. ICML, 2020
Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*

NOISE MITIGATION TECHNIQUES: EFFICACY OF ROBUST LEARNING TECHNIQUES

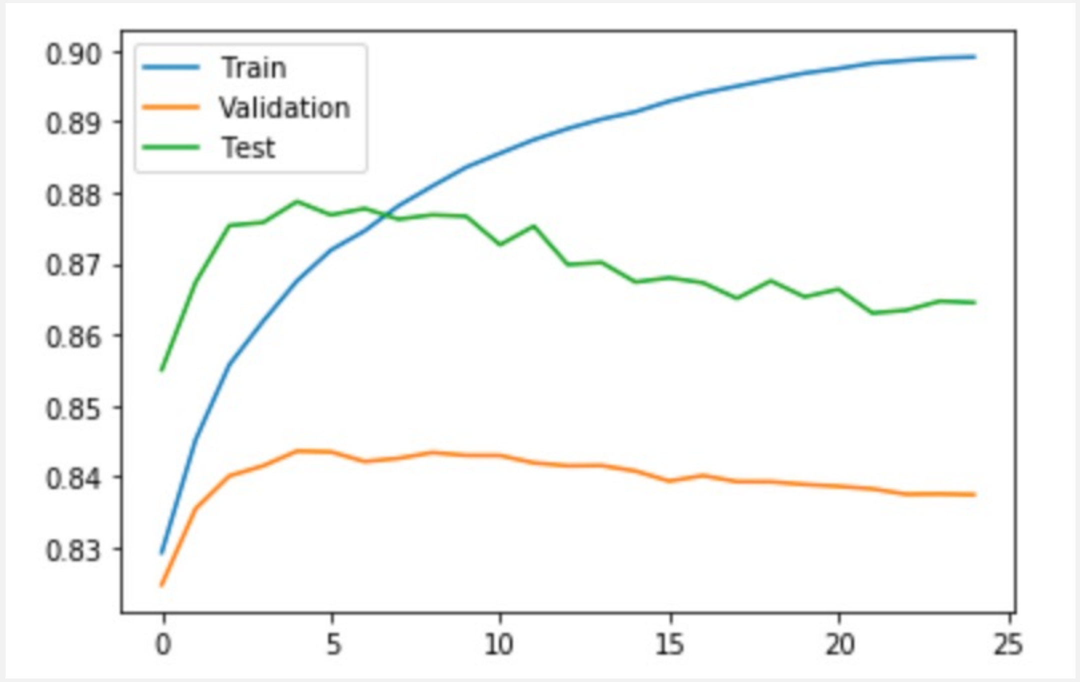
- We find that Early Stopping works as well as any other technique!
- Further, other techniques needs careful hyperparameter tuning, which early stopping does not.

			RANDOM Noise		CSIM Noise	
Dataset	Model	Clean	10%	50%	10%	50%
ATIS	BERT	0.9782	0.9317	0.6443	0.9421	0.5935
	RobustLoss	**0.9753**	**0.9727**	0.9044	0.9399	0.8955
	Noise Layer	0.9417	0.9238	0.8842	0.9025	0.8383
	LIMIT	0.9428	0.9484	0.8936	0.9428	**0.9148**
	Early Stopping	0.9742	0.9700	**0.9053**	**0.9673**	0.9037
SNIPS	BERT	**0.9836**	0.9371	0.5866	0.9481	0.6062
	RobustLoss	0.9814	**0.9752**	0.9468	**0.9757**	0.9142
	Noise Layer	0.9642	0.9733	0.9576	0.9652	0.9400
	LIMIT	0.9628	0.9628	0.9628	0.9628	**0.9685**
	Early Stopping	0.9757	0.9717	**0.9657**	0.9692	0.9540
AGNews	BERT	0.9364	0.8934	0.5334	0.8948	0.5373
	RobustLoss	0.9296	0.9228	0.8961	0.9222	0.8587
	Noise Layer	0.9252	0.9284	0.8889	0.9265	0.8595
	LIMIT	0.9034	0.9017	0.8806	0.9002	0.8288
	Early Stopping	**0.9368**	**0.9310**	**0.9060**	**0.9325**	**0.8843**

Anoop Kumar, Pankaj Kumar Sharma, Aravind Illa, Pritam Varma, Abhinash Khare, Salil Aggarwal Anurag Dwarakanath, Sriram Venkatapathy, Vinod Mamtani, Aram Galstyan, "Effectiveness of Early Stopping for Text Classification with Noisy Labels", under submission at EMNLP 2021

NOISE MITIGATION TECHNIQUES: EFFICACY OF ROBUST LEARNING TECHNIQUES

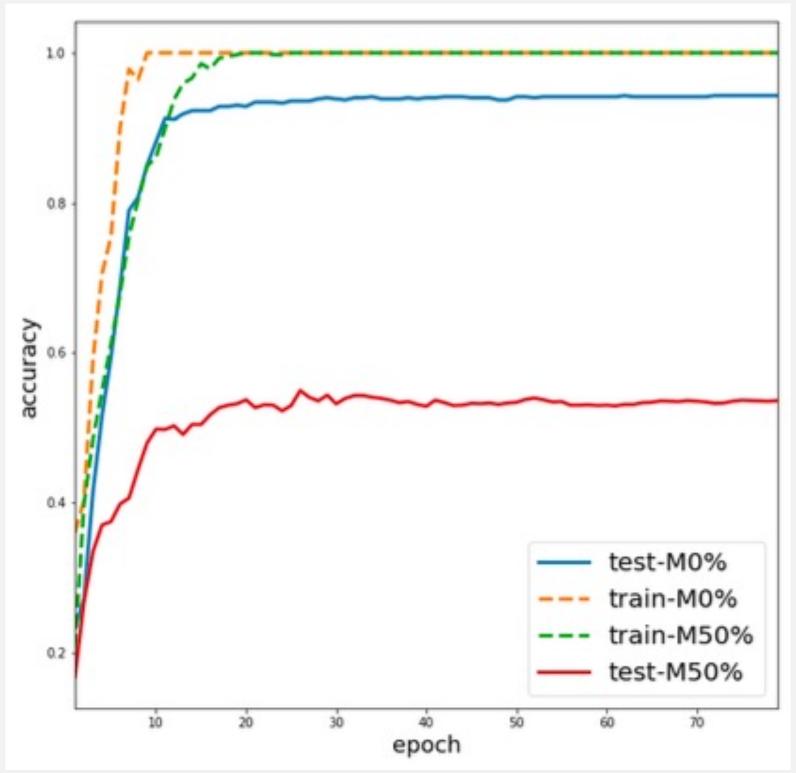
- Early Stopping works equivalently well across datasets & noise models
- This can be visualized in the validation curve.



Training and Validation curves on a noisy dataset

NOISE MITIGATION IN THE LOW DATA REGIME

- Can early stopping maintain performance even in the low data regime case?
  - When we have very little supervised data to train on.
- Results show that all methods including Early Stopping cannot work in the low data regime.



Anoop Kumar, Pankaj Kumar Sharma, Aravind Illa, Pritam Varma, Abhinash Khare, Salil Aggarwal Anurag Dwarakanath, Sriram Venkatapathy, Vinod Mamtani, Aram Galstyan, "Effectiveness of Early Stopping for Text Classification with Noisy Labels", under submission at EMNLP 2021

CONCLUDING REMARKS

- SLU methods have made significant progress in recent years.
- In the Indic region, various challenges exists:
  - Tackling Transliterated Input
  - Expanding language capability
  - Being Robust to noise in data.