Beyond Fuzzy Matching: Effective ways to Transfer Learning in NLP

Sruthi Poddutur
poddutur@stanford.edu

https://youtu.be/-Dt0SpJmOzQ

1.Problem and work:

- For data integrations from different sources, given a database of organizations, perform fuzzy matching to fetch most relevant org record in DB
- Study the nature of Transfer Learning (**TL**) in NLP using transformer architectures (BERT etc)
- Differentiate TL in NLP vs Image processing
- BaseLine: SVM model with 4 edit-distance features
- Explored two different types of TL inNLP:
  *Feature encoder Based
  *Fine-Tuning Based
  - Vanilla: Unfreeze Bert & tune weights
  - Improving Para Efficiency: Trimmed Models
  - Differential Learning Rate
  - Contrastive Loss based

2.DataSet:

Crawled Wikidata and DBPedia SPARQL Queries to fetch OrgNames and aliases
TrainingSize: 20,000 and **TestSize**: 5,000

OrgName1	OrgName2
IBM	International Business Machines
High Performance Alloys	HP Alloy
IAC	Intl Automotive Components
7 Eleven	Seven Eleven
Tyson foods	Tyson Mexican Original
Make and Mold	Make n Mold
Ratliff Readymix	Ratliff Ready Mix

3.Novel Findings:

- Beyond Fuzzy Match: NLP-TL has knowledge of complex relations like ‘Acquisition’ (PepsiCo & QuakerOats in 2001), ‘Family Tree’ (CharityInc & AIG), ‘Renamed’(CMPInfo,UBMIntermediate)
- NLP-TL vs Image-TL: CNNs have a hierarchical structure for the knowledge learnt going from generic layers to task-specific last layers which can be tapped for varied use. Similar to CNNs, pre-trained Bert has huge knowledge bank, but it has no hierarchical structural knowledge. NLP TL on-boards user with both needed knowledge as well as much bigger unwanted knowledge for the task.
- Pre-process block: NLP TL with lack of control on selecting specific knowledge is forcing the need for pre-blocking to removing huge unwanted knowledge like ‘Competitors’, ‘Same Industry’, ‘Shared-location’ for this task

Thanks: TAs esp Fatma Tlili, Atharva Parulekar and Instructors

4.Other Findings:

- **BaseLine Model**: RBF SVM with 4 edit dist features gave 85% train & test
- Feature Based TL is sensitive to InputFormat and EncodingScheme: <CLS>OrgNAme1<SEP>OrgNAme2<SEP> input format with Mean of the last 3 layers encoding scheme gave the best result
- If right InputFormat and Encoding scheme are not used then Feature-Based TL could be out beaten by simple baseline model with barely 4 features
- FineTune TL has faster convergence and has more learning ability
- FineTune based model could be trimmed to half the size without hurting performance
- Differential learning rate based fine tuning did not help much in performance
- Contrastive loss was used to try and learn true embeddings to entity. But, it is observed that without harnessing much harder triplets it might not be good at this task
- FineTuning with vocab is hard: Bert being pre-trained to ignore1k unknown tokens in the vocab, finetuning with the new 1k vocab needs lot of data and epochs to see any good numbers.
- FineTuning with layers is simple: Takes less data and just few (<10) epochs and is something the work highly recommends for good accuracy

5.FeatureBased TL Models:

Sensitive to input format and encoding schema

DNN1 (Layers)	Output Shape
CONCAT(Bert-Embedding(OrgName1), Bert-Embedding(OrgName2))	(None, 768)
Dense(10)	(None, 10)
Dense(1)	(None,1)

DNN2 (Layers)	Output Shape
CONCAT(Bert-Embedding(OrgName1), Bert-Embedding(OrgName2))	(None, 768)
Dense(256), Dropout(0.1)	(None, 256)
Dense(10), Dropout(0.1)	(None, 10)
Dense(1)	(None,1)

DNN3 (Layers)	Output Shape
CONCAT(Bert-Embedding(OrgName1), Bert-Embedding(OrgName2))	(None, 768)
Dense(256), Dropout(0.2)	(None, 256)
Dense(10)	(None, 10)
Dense(1)	(None,1)

DNN4 (Layers)	Output Shape
Bert‘s Last Encoder Layer Output for input: $\langle CLS\rangle\, OrgName1\, \langle SEP\rangle\, OrgName2\, \langle SEP\rangle$	(None, 768)
Dense(256),Dropout(0.2)	(None, 256)
Dense(10)	(None, 10)
Dense(1)	(None,1)

DNN5 (Layers)	Output Shape
Mean of Bert‘s last 3 Encoder Layers Output for input: $\langle CLS\rangle\, OrgName1\, \langle SEP\rangle\, OrgName2\, \langle SEP\rangle$	(None, 768)
Dense(256),Dropout(0.2)	(None, 256)
Dense(10)	(None, 10)
Dense(1)	(None,1)

Model	Iterations to Converge	Train Acc	Val Acc
DNN1	6	80.4%	78.9%
DNN2	32	92.5%	79.9%
DNN3	19	85.3%	80.2%
DNN4	18	70.05%	68.30%
DNN5	11	91.15%	85.04%

6.FineTuneBased TL Models:

Finetuning converges faster(only 3epochs) & has more learning ability
Model could be trimmed to HALF without hurting performance

Model Name	Trimmed To	Num Bert’s Encoder Layers Used	Architecture	Iterations With Early Stopping	Train Acc	Val Acc
DNN6	FullSize	12	DNN5+Bert 12th Layer Unfrozen	3	92.1%	90.06%
DNN7	1/12th	1	DNN5+Bert 1st Layer Unfrozen	3	91.39%	81.42%
DNN8	2/12th	2	DNN5+Bert 2nd Layer Unfrozen	1	91.88%	80.72%
DNN9	3/12th	3	DNN5+Bert 3rd Layer Unfrozen	3	94.83%	86.47%
DNN10	4/12th	4	DNN5+Bert 4rth Layer Unfrozen	3	95.61%	86.67%
DNN11	5/12th	5	DNN5+Bert 5th Layer Unfrozen	3	94.91%	88.53%
DNN12	6/12th	6	DNN5+Bert 6th Layer Unfrozen	11	91.06%	89.53%

7.Some Interesting Results:

Wanted (VS) Unwanted Knowledge from Pretrained BERT TL

Interesting Correct predictions by DNN6

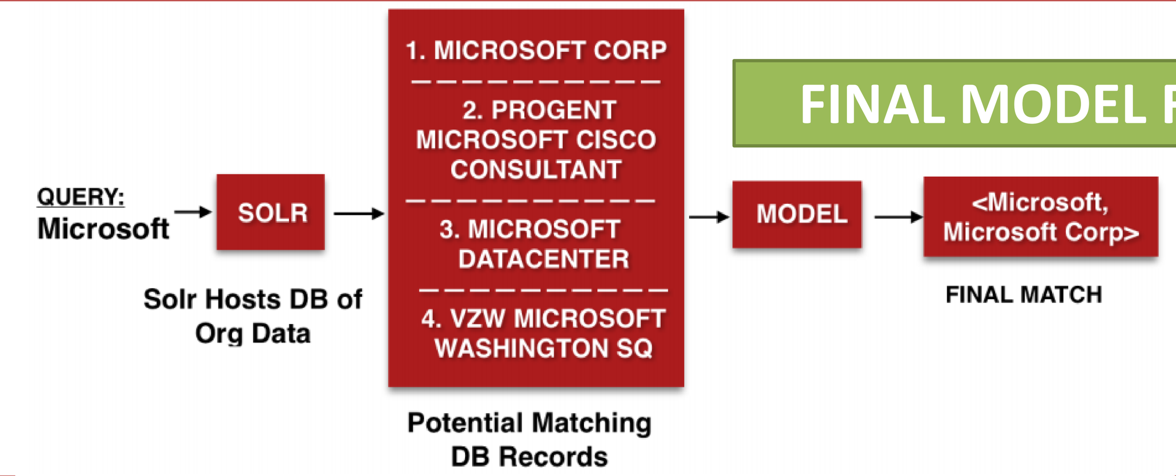
OrgName1	OrgName2	Relation
Quaker Oats	PepsiCo	Aquired (2001)
US Oncology	McKesson	Aquired (2010)
LourdesMedicalCenter	Trinity Health	Same Family
Tamoil	LibyanNationOil	Same Family
IBM	Big Blue	Alias
TSG Consulting	Transportes Sousa Gomes	Abbreviation
CMP INFO	UBM Intermediate	Org Rename
Chartis	AIG	Org Rename

Interesting Wrong Predictions by DNN6
(SameIndustry or CoLocated = SICL)

OrgName1	OrgName2	Relation
Penn State Univ	Kent State Univ	SICL
Univ of Iowa	Univ of Utah	SICL
Polytechnic Inst NYU	Worcester Polytechnic Inst	SICL
Towa Bank	United Western Bank	Same Industry
Amazon	Microsoft	Competitors
Airtel	Vodafone	Competitors

Unlike CNN TL, where user has control to pick specific wanted features/knowledge,NLP TL doesnt give control. Its either all in or not

Pre-Blocking: Proposed fix to counteract Unwanted learnings



FINAL MODEL FOR THE TASK

8.Differential Learning Rate(LR) FineTuning:

- DNN6 architecture is taken
  - BERT layers were assigned lower 1e-5 LR
  - Dense layers were assigned 1e-3 LR (same as DNN6)
- **Results match DNN6**: 3 epochs, obtained 92.09% train & 89.49% test acc
- **Analysis**: Higher the data correlation between pre-trained (Bert MLM task) model & our task model, lesser the gap in LRs between these layers

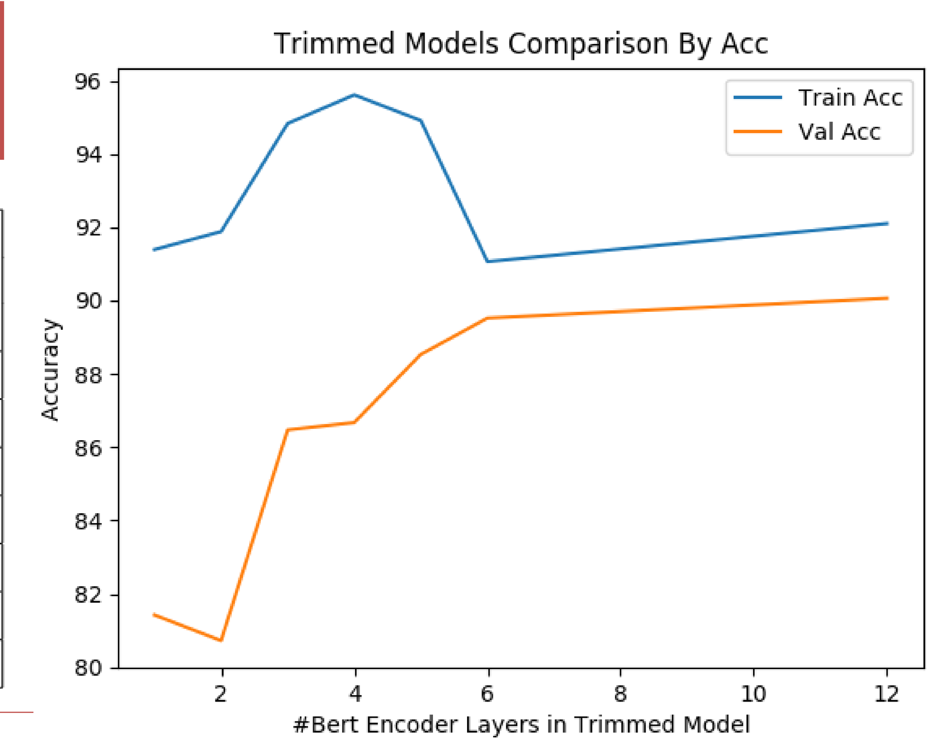
9.Contrastive Learning Fine Tuning:

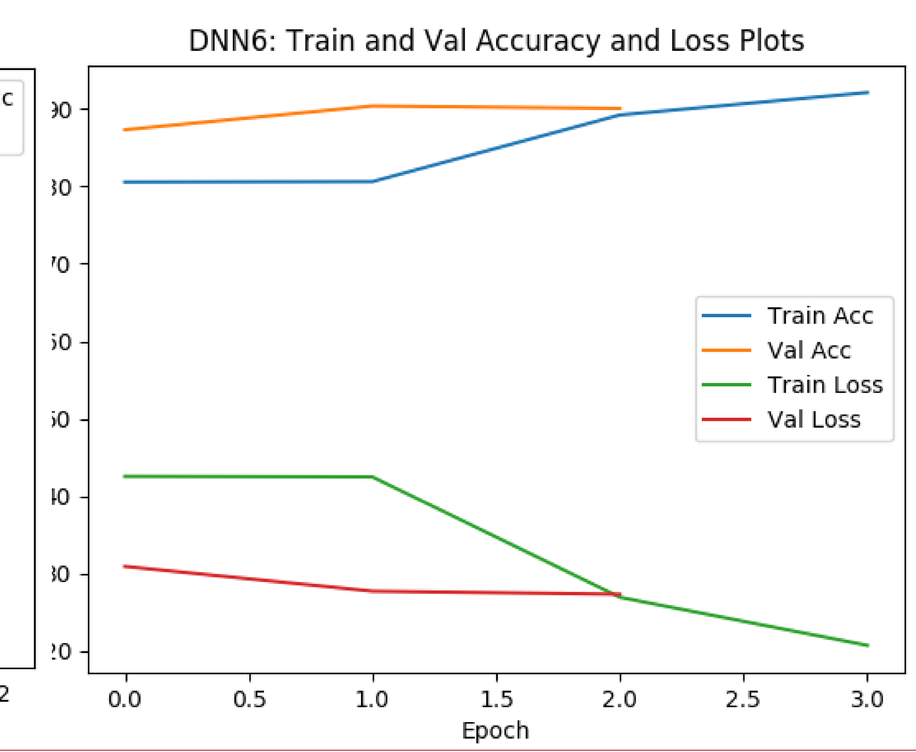
Need harder pairs for better acc’s

Contrastive Loss DNN14 (Layers)
emb1=Bert-Last-Enc-Layer-Embedding(OrgName1)
emb2=Bert-Last-Enc-Layer-Embedding(OrgName2)
ContrastiveLoss(Euclidean-Distance(emb1, emb2))

Contrastive Loss DNN15 (Layers)
emb1=Bert-Last-3Enc-Layers-Embedding-Mean(OrgName1)
emb2=Bert-Last-3Enc-Layers-Embedding-Mean(OrgName2)
ContrastiveLoss(Euclidean-Distance(emb1, emb2))

- DNN14 saturated at 74% while DNN15 could barely give ~54%
- **Error Analysis**: It couldnt learn harder pairs e.g.,alias-pairs <BigBlue,IBM>, Harder Abbrevs <SouthGeorgiaCottonLLc,SO GA Cotton WholeSale>





References:(1) J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert:Pre-training of deep bidirectional transformers for language understanding, 2018
(2) A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017
(3) M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations
(4) Hierarchical entity matching using bi-grus https://www.microsoft.com/en-us/research/uploads/prod/2019/04/auto-em.pdf