Beyond Fuzzy Matching: Effective ways to Transfer Learning in NLP

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 in NLP: *Feature encoder Based* and *Fine-Tuning Based*.
- Vanilla: Unfreeze Bert & tune weights.
- Fine-Tuning Based: RBF SVM with 4 edit dist features gave 85% train & test performance.
- 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 performance.
- 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 ignore 1k 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.

2. Dataset:
- Crawled Wikidata and DBPedia SPARQL Queries to fetch orgNames and aliases.
- TrainingSize: 20,000 and TestSize: 5,000.

<table>
<thead>
<tr>
<th>OrgName1</th>
<th>OrgName2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>HP Alley</td>
</tr>
<tr>
<td>High Performance Alloy</td>
<td>Int'l Automotive Components</td>
</tr>
<tr>
<td>7 Eleven</td>
<td>Seven Eleven</td>
</tr>
<tr>
<td>Tyson foods</td>
<td>Tyson Mexican Original</td>
</tr>
<tr>
<td>Make and Mold</td>
<td>Make n Mold</td>
</tr>
<tr>
<td>Ratliff Realmix</td>
<td>Ratliff Ready Mix</td>
</tr>
</tbody>
</table>

3. Novel findings:
- Beyond Match: NLP-TL has knowledge of complex relations like ‘Acquisition’ (PepsiCo & QuakerOats in 2001), ‘Family Tree’ (CharityLink & AIG), ‘Renamed’ (CMPInfo, UBMIntermediate).
- NLP-TL vs Image-TL: CNNs have a hierarchical structure for the knowledge learnt from generic layers to task-specific last layers.
- 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.
- 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.

4. Other findings:
- **Baseline Model**: RBF SVM with 4 edit dist features gave 85% train & test performance. 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 ignore 1k 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. Feature Based TL Models:
- **Sensitivies to input format and encoding schema**
  - DNN5 (Layers): Output Shape
    - CONCASC-Bert-Embedding (OrgName1), Bert-Embedding (OrgName2)
    - Dense(256), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
  - DNN6 (Layers) Output Shape
    - CONCASC-Bert-Embedding (OrgName1), Bert-Embedding (OrgName2)
    - Dense(256), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
  - DNN7 (Layers): Output Shape
    - Bert's Last Encoder Layer Output for input: (CLS) OrgName1<SEP>OrgName2<SEP> Output Shape
    - Dense(256), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
  - DNN8 (Layers): Output Shape
    - CONCASC-Bert-Embedding (OrgName1), Bert-Embedding (OrgName2)
    - Dense(256), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)
    - Dense(10), DropOut(0.1)

6. Fine Tune Based TL Models:
- **Finetuning converges faster (only 3 epochs)** & has more learning ability.
- Model could be trimmed to HALF without hurting performance.

7. Some interesting results:
- **Wanted (VS) Unwanted Knowledge from Pretrained BERT TL**
  - **Interesting Correct predictions by DNN6**
    - Quaker Oats | OrgName1 | OrgName2 | Relation | Aquired (2003) | OrgName3 | OrgName4 | Relation |
    - PepsiCo | OrgName5 | OrgName6 | OrgName7 | OrgName8 | OrgName9 | OrgName10 | OrgName11 |
  - **Interesting Wrong Predictions by DNN6**
    - SameIndustry or CoLocated = SSLC (SameIndustry or CoLocated = SSLC)
    - Univ of Iowa | OrgName13 | OrgName14 | Relation | Polosukhin | OrgName15 | OrgName16 | Relation |
    - Univ of Utah | OrgName17 | OrgName18 | Relation | Uszkoreit | OrgName19 | OrgName20 | Relation |
  - **BaseLine**
    - **TL has faster convergence and has more learning ability**
    - **Differential Learning Rate**
      - **Contrastive Loss based**
        - Differential Learning Rate based fine tuning did not help much in performance.
        - **Differentiate TL in NLP vs Image processing**
          - **Analysis**
            - Differential Learning Rate based fine tuning did not help much in performance.
            - **Different Learning Rate (LR) Fine Tuning**
              - DNN 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 & our task model, lesser the gap in LRs between these layers.
              - **9. Contrastive Learning Fine Tuning**
                - Need harder pairs for better acc's
                - DNN14 saturated at 74% while DNN15 could barely give ~54% acc
                - Error Analysis: It couldnt learn harder pairs e.g., alias-pairs, BigBlue,IBM, Harder Abbrevs SouthGeorgiaCotton LLC, CGA Cotton WholeSale

8. Differential Learning Rate (LR) Fine Tuning:
- **DNN 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 & our task model, lesser the gap in LRs between these layers.

9. Contrastive Learning Fine Tuning:
- Need harder pairs for better acc's
- 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 SouthGeorgiaCotton LLC, CGA Cotton WholeSale

References: