Unsupervised Domain Clusters in Pretrained Language Models

ACL 2020

Roee Aharoni[1] **& Yoav Goldberg**[1,2]
[1] Computer Science Department, Bar Ilan University
[2] Allen Institute for Artificial Intelligence

Overview

- **Understand Clustering Properties of Pre-trained Language Models**
- **In-domain Data Augmentation using pre-trained embeddings**
  - Distance-based Retrieval
  - Binary Classification Fine-tuning
- **Application to Neural Machine Translation**

Motivation

- High quality data is a key aspect in training ML models
- Real-world NLP problems -> we may not have access to sufficient *in-domain* labelled data
- Massive pre-trained models -> great progress on many NLP benchmarks
- **How can we make use of the nice properties of pre-trained models like BERT to augment our in-domain data?**

Preliminary Experiments

Pilot Study - Dataset

- Textual data in five diverse domains:
  - Movie subtitles
  - Medical text
  - Legal text
  - Translations of the Koran
  - IT-related text

- Sample 2000 distinct sentences from each domain -> cluster embeddings

- Here, different topics are referred to different domains

PCA Visualization

- Massive pre-trained LMs implicitly learn sentence representations that cluster by domains without supervision
- Utilize this property for data augmentation

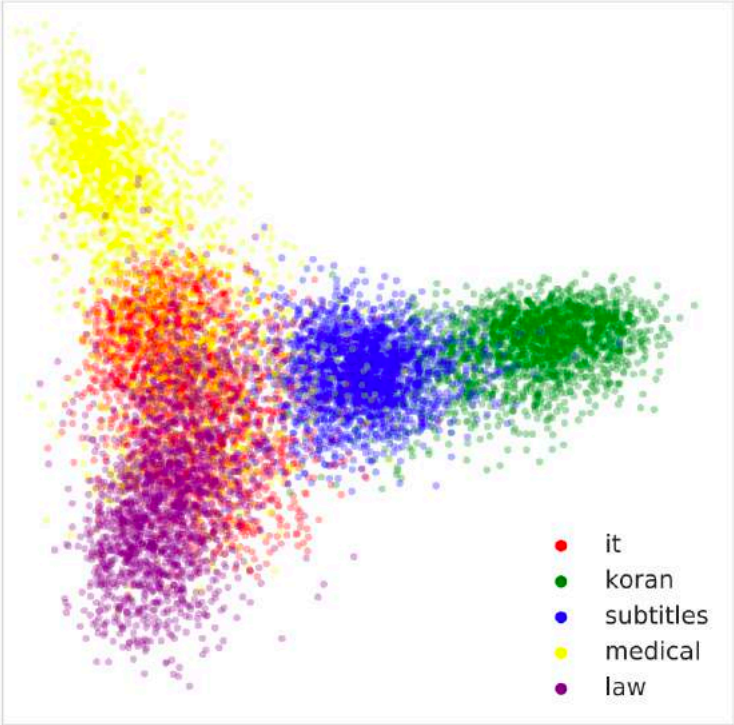

Figure 1: A 2D visualization of average-pooled BERT hidden-state sentence representations using PCA. The colors represent the domain for each sentence.

Clustering

Sentence vectors

↓

PCA

↓

Gaussian Mixture Models
GMM (k=5, 10, 15)

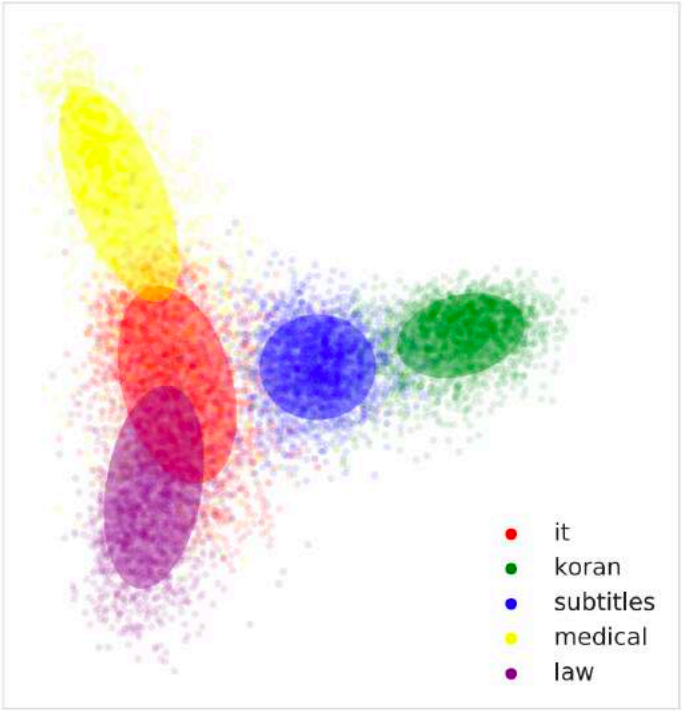

Figure 2: A 2D visualization of the unsupervised GMM clustering for the same sentences as in Figure

Quantifying the Clustering Property

- Need a quantitative way to evaluate the “goodness” of the resulting clusters
- Note that in these experiments, we have true labels
- **Purity Metric**:
  - Each cluster is assumed to have the label corresponding to the most common class of the sentences in that cluster
  - Compute the *accuracy* according to this majority-based assignment.

Quantifying the Clustering Property

	k=5	k=10	k=15
Random	15.08 (±0.0)	16.77 (±0.0)	17.78 (±0.0)
LDA	24.31 (±0.99)	26.73 (±2.19)	30.79 (±2.97)

	with PCA (n=50)			without PCA		
	k=5	k=10	k=15	k=5	k=10	k=15
word2vec	53.65 (±0.79)	68.14 (±2.58)	73.44 (±0.68)	45.93	65.80	76.26
BERT-base	**87.66 (±0.24)**	88.02 (±1.10)	88.37 (±0.66)	**85.74**	85.08	86.37
BERT-large	85.64 (±6.13)	87.61 (±0.26)	89.07 (±0.53)	68.56	**86.53**	86.99
DistillBERT	83.68 (±7.14)	86.31 (±0.86)	87.53 (±0.85)	79.00	86.42	**88.14**
RoBERTa-base	79.05 (±0.10)	86.39 (±0.90)	86.51 (±0.28)	70.21	80.35	81.49
RoBERTa-large	80.61 (±0.33)	**89.04 (±0.15)**	**89.94 (±0.23)**	69.88	81.07	85.91
GPT-2	70.30 (±0.05)	84.76 (±0.30)	82.56 (±1.29)	37.82	39.02	41.45
XLNet	55.72 (±0.69)	68.17 (±3.93)	72.65 (±1.92)	30.36	32.96	48.55

Table 1: Unsupervised domain clustering as measured by purity for the different models. Best results are marked in bold for each setting.

- MLM-based models dominate
- Reason: MLM-based models use the entire sentence context
  - while the auto-regressive models only use the past context and word2vec uses a limited window context
- Using PCA improved performance in most cases

Analyzing Incorrect Assignments

Subtitles assigned to IT
Push it up to the front of the screen.
Polyalloy requires programming to take permanent form.
Law assigned to Medical
- Viruses and virus-like organisms
where the glucose content is equal to or less than the fructose content.
Medical assigned to Law
This will be introduced by a Regulation adopted by the European Commission.
The marketing authorisation was renewed on 22 May 2002 and 22 May 2007.
IT assigned to Medical
R65: Harmful: may cause lung damage if swallowed
Automatic Red-Eye Removal

- Some of the mis-assignments make sense
- Usually shorter —> maybe due to the lack of sufficient contextual information

Cross-Domain NMT Experiments

- **Without** domain data selection

- Train domain-specific models for each of the domains
- Evaluate each model across the different domain test sets,

- Understand the effect of training with different domains on the downstream MT performance

- Transformer Encoder Decoder (Vaswani et. al, 2017)

Cross-Domain NMT Results

	Medical	Law	Koran	IT	Subtitles
Medical	**56.5**	18.3	1.9	11.4	4.3
Law	21.7	**59**	2.7	13.1	5.4
Koran	0.1	0.2	15.9	0.2	0.5
IT	14.9	9.6	2.8	**43**	8.6
Subtitles	7.9	5.5	6.4	8.5	27.3
All	53.3	57.2	**20.9**	42.1	**27.6**

Table 4: SacreBLEU (Post, 2018) scores of our baseline systems on the test sets of the new data split. Each row represents the results from one model on each test set. The best result in each column is marked in bold.

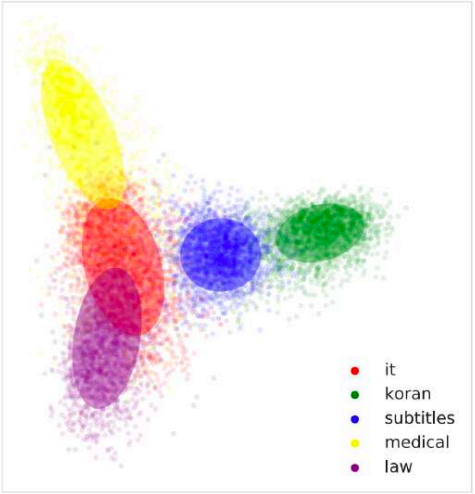

Figure 2: A 2D visualization of the unsupervised GMM clustering for the same sentences as in Figure

- Trained on IT: Tested on Medical vs. **Tested on Koran**
- Preliminary visual analysis can be a useful tool for understanding the relationship between diverse datasets (compatible domains)

Cross-Domain NMT Results

	Medical	Law	Koran	IT	Subtitles
Medical	**56.5**	18.3	1.9	11.4	4.3
Law	21.7	**59**	2.7	13.1	5.4
Koran	0.1	0.2	15.9	0.2	0.5
IT	14.9	9.6	2.8	**43**	8.6
Subtitles	7.9	5.5	6.4	8.5	27.3
All	53.3	57.2	**20.9**	42.1	**27.6**

Table 4: SacreBLEU (Post, 2018) scores of our baseline systems on the test sets of the new data split. Each row represents the results from one model on each test set. The best result in each column is marked in bold.

- In-domain training data is best for each domain (even better than using all-available data)
- **Using the right data is critical for achieving good performance on an in-domain test set, and more data is not necessarily better**
- Koran test set: Training on all the available data helped (Because training data size was considerably small for this domain)

Domain Data Selection

Domain Data Selection

- Task of selecting the most appropriate data for a domain from a large corpus, given a smaller set (~2000) of in-domain data

- Use cases:
  - Train a domain-specific model from scratch
  - Fine-tune a pre-trained general-domain model
  - Prioritize data for annotation as in an Active-Learning framework

Method 1: Distance Based Retrieval

- First compute a **query** vector
  - Element-wise average over the vector representations of all the sentences in the small in-domain set (~2000).

- Retrieve the most relevant sentences from the large general-domain training set
  - By computing the cosine similarity of each sentence with the query vector and ranking the sentences accordingly
  - Pick top-K sentences to augment the in-domain dataset

Method 2: Binary Classification Fine-Tuning

- Fine- tune the pretrained LM (e.g. BERT) for binary classification:
  - Use the in-domain sentences as positive examples
  - Randomly sampled sentences (from a subset of rest of unlabelled data) as negative examples
  - For dataset augmentation, apply this classifier on the general-domain data set and pick the sentences that are classified as positive as in-domain
  - or choose the top-K sentences as ranked by the classifier output probability.

- **Negative Sampling with Pre-ranking**
  - Problem: random negative samples deteriorate the classifier performance.
  - Instead, perform a biased sampling of negative examples.
    - First rank the general-domain data using the Domain-Cosine (Method-1), and then sample negative examples from the bottom two-thirds.
  - Classifier obtains better precision

Baseline (Moore and Lewis, 2010)

- For each candidate sentence, compute:
  - L1: the log-likelihood according to a domain-specific language model,
  - L2: the log-likelihood a non-domain-specific (general) language model
  - Ranked sentences by L1-L2 the difference in log-likelihood
- Pick top-K candidate sentences

- It is based on simple n-gram language models
  - Cannot generalize beyond the n-grams that are seen in the in-domain set.
  - In addition, it is restricted to the in-domain and general-domain datasets it is trained on, which are usually small.
  - On the contrary, pre-trained LMs are trained on massive amounts of text

Experiments

- 2000 in-domain sentences from each domain.
- For the general-domain corpus, concatenate the training data from all domains ~1.4M
- On NMT task, compare their model to 4 approaches:
  1. Moore and Lewis (2010) Baseline
  2. A random selection baseline
  3. An oracle which is trained on all the available in-domain data (access to true labels)
  4. The model trained on all the domains concatenated

Results

	Medical	Law	Koran	IT	Subtitles	Average
Random-500k	49.8	53.3	18.5	37.5	25.5	36.92
Moore-Lewis-Top-500k	55	58	21.4	42.7	27.3	40.88
Domain-Cosine-Top-500k	52.7	58	**22**	42.5	27.1	40.46
Domain-Finetune-Top-500k	54.8	58.8	21.8	**43.5**	27.4	**41.26**
Domain-Finetune-Positive	55.3	58.7	19.2	42.5	27	40.54
Oracle	**56.5**	**59**	15.9	43	27.3	40.34
All	53.3	57.2	20.9	42.1	**27.6**	40.22

Table 6: SacreBLEU scores for the data selection experiments. Highest scores are marked in bold.

- Results are appealing given that only 2000 in-domain sentences were used for selection for each domain out of 1.45 million sentences.

Conclusions

Summary

- Clustering Properties of Pre-trained Language Models

- In-domain Data Augmentation using pre-trained embeddings
  - Distance-based Retrieval
  - Binary Classification Fine-tuning

- Application to Neural Machine Translation
- **For text classification**: Rather than simply using manually curated keyword list to identify similar text for augmenting training data, use pre-trained LM embeddings

- Would this work for paragraph / document level embeddings ?

Thank You

Questions?