Team 5: Acronym Extraction and Disambiguation

Snehal Kumar, 2019101003
Shrey Gupta, 2019101058
K V Aditya Srivatsa, 2018114018
Mukund Choudhary, 2018114015

1. Link to Project & Video

The following link has READMEs wherever needed, datasets (and variations) used, results, analysis files, and implementations: Github Repo

Video Presentation: Video

Note: since it is a relatively new task thus we have briefed on the task and the dataset too (apart from related work).

2. Introduction & Related work

Acronyms an important part of processing text in most domains, more prominently so in technical domains. But very few models and datasets actually address this issue exhaustively. Most of these are from a developing Shared Task [6] which boasts of the biggest corpus for tasks like Acronym Extraction and Disambiguation (as described below) across two major domains, Scientific & Legal.

2.1. Acronym Extraction

Most dictionaries of acronyms and their associated long forms are not exhaustive and need to be updated regularly. Even after these measures, a system can be fooled by new acronyms or missing period marks etc. The task of Identification of an acronym and Extracting them is difficult and important. Thus, in this subtask, we aim to identify short forms and the related long forms from the documents in the dataset given.

2.2. Acronym Disambiguation

Similarly, a lot of acronyms can be expanded to multiple different long forms, a popular example among college students is how M.I.T. could either be expanded with Manipur at the start or with Massachusetts. Thus Acronym Disambiguation is another important aspect of this task. Thus in this subtask, we aim to select the correct meaning from a set of possible meanings of an ambiguous acronym in a given sentence of the dataset.

2.3. Related Work

Some of the related work is summarised as below:

- BERT-based Acronym Disambiguation with Multiple Training Strategies[5]: For the disambiguation task, they use the pretrained scibert-scivocab-uncased model. There best performing model makes use of psuedo-labelling [3][4], which considers model predictions having high confidence as ground truth labels. This pseudo-labeled data is mixed with the original training data, and the augmented data is then re-used.

- Primer AI's Systems for Acronym Identification and Disambiguation[1]: **For identification**, they use XLNet-Large to generate the token level embeddings of the input text, which are passed through a linear layer to generate a size-5 output vector of BIO tags. The label of a token is based on the maximum logit score. **For disambiguation**, they consider it as an information retrieval task where given a test sentence containing an acronym, they try to identify the most similar training sentence seen before and use its labels.

- AT-BERT: Adversarial Training BERT for Acronym Identification Winning Solution for SDU@AAAI-21[7]: For the identification task, they use FGSM[2] driven adversarial training of a linear layer head over a base encoder. This approach generates new samples by specifically modifying existing embeddings in the direction of increase of the gradient preventing overfitting despite the small corpus. They use an ensemble of many such pipelines, with various base encoders (BERT, SciBERT, RoBERTa, ALBERT, ELECTRA).

3. Dataset Description

The dataset is uniform overall and uses standard BIO format whenever needed. All data is also in regular JSON format, thus no non-linguistic cleaning is needed.

3.1. Acronym Extraction

The dataset consists of English sentences of scientific and legal domains (among other data). It has 4000 paragraphs each in both domains and can be viewed here.

3.2. Acronym Disambiguation

The dataset consists of English sentences of scientific and legal domains (among other data). It has 457 acronyms in the Scientific domain & 273 in the Legal one and can be viewed here.

3.3. Preliminary Stats/Baseline

Above are the results on English data from a rudimentary rule-based system designed for the task (amount of datapoints mentioned in brackets, these can contain none, one, or multiple acronyms).

Task AE (8486)	**Precision**	**Recall**	**F1**
Scientific (4477)			
Train Set (3980)	34.02	14.88	20.71
Dev Set (497)	33.17	14.52	20.2
Legal (4009)			
Train Set (3564)	33.55	9.93	15.32
Dev Set (445)	32.34	10.24	15.56
Task AD (11760)	**Precision**	**Recall**	**F1**
Scientific (8426)			
Train Set (7532)	70.35	32.13	44.11
Dev Set (894)	71.36	33.81	45.88
Legal (3334)			
Train Set (2949)	62.51	42.18	50.37
Dev Set (385)	55.81	35.79	43.61

4. Datasets & Experiments

Post the mid-checkpoint, several experiments were performed over the respective datasets for the identification and disambiguation tasks. For AE, the data was cleaned for special tokens, while retaining some pivotal characters. For AD, an end-to-end cleaning using clustering was used to generate augmented variants of the original dataset.

4.1. AE trials

4.1.1 Mid Phase

We tried the above mentioned crude rule based system and the following:

We started with a pretrained BERT encoder with a Multi-Layer Perceptron Head. For this, we used the BERT-Sklearn library's BertTokenClassfier class. The last MLP head is set to linear activation, with an output dimensionality of 3 (for the three BIO-less labels) and class predictions are made using a softmax layer. The model was trained for 3 epochs. The architecture of the underlying BERT Model and the rest of the training parameters are similar to the architecture and training parameters used in the Final Models in the End-Phase trials and thus can be referred to from section 5.1.

4.1.2 End Phase

After the above trials, we found out that most model preformances suffered due to special tokens, but parentheses were pivotal (more in the Analysis section). Moreover, we wanted to see if the original BERT model did better than SciBERT on the legal dataset; Trials for all of these are described below.

We experimented with several variations of the dataset by removing special tokens which consisted of specific symbols, numbers and both, the results of a few of which have been reported in section 6.1. We used BERT and SciBERT in these trials for the legal data however only SciBert was used for the Scientific Data as it has been specifically been finetuned for the domain.

The number of trainable model parameters were another point of exploration. The original baseline uses a simple linear layer above the encoder embeddings. We experimented by adding more MLP layers on top as well as by changing the layer sizes. We found that increasing the number of layers does help model performance initially, but quickly causes overfitting beyond one hidden layer due to the limited amount of data. Thus, the model layers and dimensionality were iteratively increased as per validation performance, and restricted as soon as performance dropped.

4.2. AD trials

Most of the experimenting was done by the Mid-Phase for this task, the same is described below.

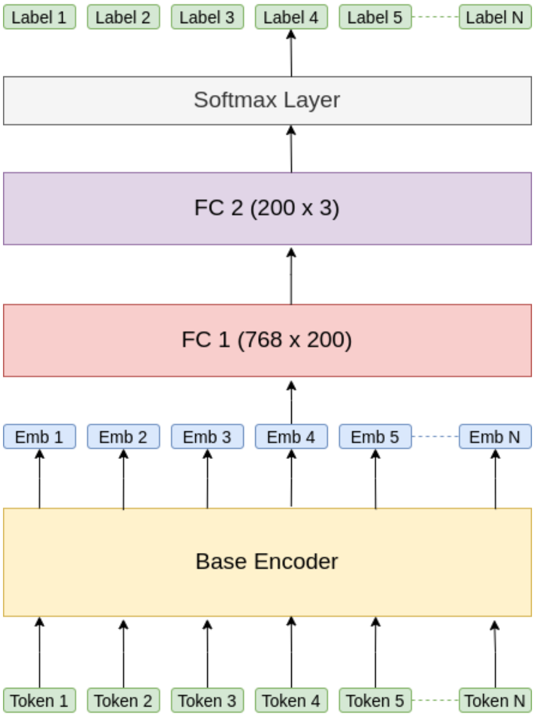
We tried the above mentioned crude rule based system and the following:

We make use of the token embeddings of a pretrained Transformer Encoder. We have trained four pipelines, one each with the Encoder as "BERT base uncased", "SciBERT cased", "SciBERT uncased" and "SciBERT uncased fine tuned on WikiPedia articles". In each model, the token level embeddings of each WordPiece processed input text is passed through a linear layer, which returns a 2-size-vector for each token. The first and second cells each, of a sequence containing n tokens are concatenated to form two separate embeddings. These are then independently passed through softmax layers. The vectors therefore represent the probability of each token in the sentence to be the starting and ending point of the expanded form of the acronym respectively.

We use F1 scores and Jaccard distances as metrics to assess the correctness of this span labelling task. For training, all of the three models have 12 layers and 12 attention heads, hidden layer size of 768, and position and token embedding sizes of 512. Additionally, all encoder fully-connected-layers use GeLU activation and a dropout of 0.1. Each model was trained for 10 epochs with an initial learning rate of 1e-5 with K-fold cross validation (K=5).

5. Final Architecture

5.1. AE



AE Final Architecture

Similar to the Mid-Phase, we use the BERT-Sklearn library's BertTokenClassfier class (any base BERT encoder variant, with an MLP head on top). The MLP contains one hidden layer of size 200 with ReLU activation after each fully connected layer. Each token embedding (size 768) passes through the hidden layer (size 200), then through a linear layer (size 3) corresponding to the 3 BIO-less labels. Additionally, the best performing models make use of SciBERT (scibert-scivocab-cased), and are trained with layer-wise dropout of 0.1, initial learning rate of $1e-4$, and a train batch size of 16 for 5 epochs.

We use F1 scores as a metric of correctness of the labels predicted as described in section 6.1 in detail.

5.2. AD

Majority of the architecture remains same as that mentioned for the mid phase above with a slight change in the initial learning rate to 2e-5. The main issue (as the Mid-Phase Analysis indicated) with the task was the data quality. Thus we re-imagined the architecture to take care of such variations, as follows:

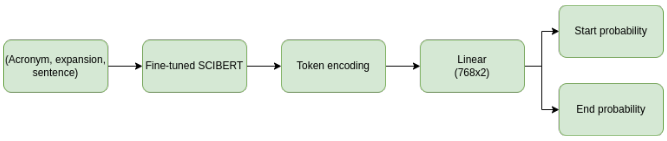
- Using diction of dataset to scrape Wikipedia articles and used them to fine-tune the SciBERT model

- Augmenting the dataset by cleaning (lowercasing and removing whitespaces)

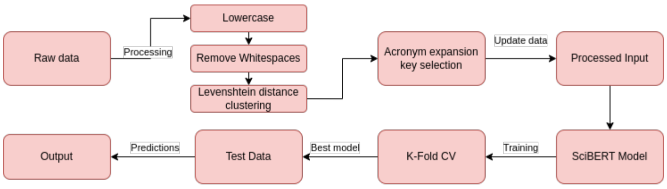
- The new concatenated expansions are used to calculate the Levenshtein distance for all pairs of expansions for an acronym. This provides a metric to cluster the similar and misspelt expansions together. Clustering is done using a threshold (40%) from the matrix generated for all pairs of expansions.

- The clustered set of expansions is used to select the most appropriate expansion as the key amongst its variations. The final acronym-key pair is used to update the training and diction files before training.

This additional step of data augmentation is done before training the model. The model is trained through 10 epochs and a 5-fold cross validation. The final predictions is given by averaging the probabilities of the trained models to get the best prediction.



AD Pipeline



AD Final Architecture

6. Evaluations mechanism & Results

6.1. AE

All the trained models are evaluated using the dev set for both legal and scientific domains of the dataset used for training. The unique representations of acronyms and long-forms are found using the predictions and the true gold labels by calculating the span of each in the token list and appending the sentence index. On the basis of these, the precision, recall and F1 score is calculated for the acronym and long-form predictions. The official evaluation metrics used are the macro-averaged precision, recall and F1 for acronym and long-form predictions which have been reported for the trials conducted and the final model.

Dataset / Model	**BERT**			**SciBert**		
(domain_flavor)	**F1**	**P**	**R**	**F1**	**P**	**R**
leg_vanilla	75.63	75.6	75.66	**75.70**	**76.10**	**75.30**
leg_nosym	70.84	71.31	70.37	72.58	73.13	72.04
leg_nosym_plusbrac	73.95	74.62	73.28	74.67	75.41	73.95
leg_nosym_plusbrac_equal	74.27	75.50	73.09	73.93	74.89	72.99
sci_vanilla				**82.74**	**81.58**	**83.94**
sci_nosym				79.35	79.3	79.4
sci_nosym_plusbrac				81.70	80.93	82.49
sci_nosym_plusbrac_equal				82.57	81.20	84.00

SciBert performs the best for both the Legal and Scientific Domains with an F1 score of **75.70** and **82.74** on the dev set of the original dataset in the BIO format.

6.2. AD

All variants of the models are evaluated on the dev set for legal and scientific domains. The actual predictions are taken from the gold labels and using the ID and label of the acronym, a match is done from the predicted file to calculate the precision, recall and F1 scores. The official evaluation metrics used are the macro-averaged precision, recall and F1 scores for the acronym expansion. The results of each evaluation have been provided.

dataset	**model**	**scibert uncased**		
	flavor	**F1**	**P**	**R**
legal	finetuned + data preproc	**89.52**	**90.92**	**88.17**
	data preproc	88.39	91.63	85.38
	finetuned	74.79	83.38	67.81
	vanilla	74.79	83.38	67.81
scientific	finetuned + data preproc	**92.33**	**94.13**	**90.6**
	data preproc	92.33	94.13	90.6
	finetuned	77.95	84.31	72.49
	vanilla	77.95	84.31	72.49

dataset	**model**	**others**		
	flavor	**F1**	**P**	**R**
legal	finetuned + data preproc			
	data preproc	88.28	90.81	85.89
	finetuned			
	vanilla	54.33	77.22	41.91
scientific	finetuned + data preproc			
	data preproc	91.25	93.74	88.88
	finetuned			
	vanilla	78.95	86.37	72.7

The values highlighted in light blue are from SciBERT cased, and the values highlighted in yellow is from Bert uncased.

7. Analysis

7.1. Dataset Quality & Baseline Findings

The annotated dataset for both tasks is well organised and labeled for a lot of parts. However we saw that for both tasks even for randomly sampled data, the dataset had blatant errors in itself, harming the result statistics a lot. For example:

- *AE*: "United Nations Population Fund", "DP", and "FPA", were marked as O (Outside of a long/short form), to name a few.
- *AD*: There were a lot of gold annotated/expected forms like: "mation", "nition", and "Noun Phras" which are misspellings or incomplete words (these examples are not misspelled but directly copied).
- A lot of random symbols, repetitions etc.

7.1.1 AE

Following are a few cases that we are working on after analysing the results from the models:

1. *Not exactly wrong*: In this case, we found:
   - *Normalised acronyms*: words like "misc" and "ad hoc" not being tagged as acronym by the model. We found that these short forms are now also used as valid words, thus explaining the BERT misunderstanding.
   - *Sequences that can be shortened*: There were sequences that could very well have an acronym but either the data didn't have it or the annotator felt that it was not an acronym in the real world, these were sometimes marked as valid long forms by the model (and not the gold data), like: "Governmental Economics and Management science Technical Institutes".
2. *Wrong*: These cases are actual mistakes where the model has marked special symbols without letters around them as "short forms" or has been affected by these symbols.

7.1.2 AD

Apart from the low quality original annotation (gold), the model prediction didn't match on the following:

1. *Extreme 1, Too vague & not too different*: These were words like "critical" which were shortened (which are also normal day to day words that we won't shorten, but a highly niche field might), for these model predicted similarly vague, but different words like "creation".

2. *Extreme 2, Too specific & too similar*: These were pairs (actual-prediction) like "Semantic Role"-"Semantic Relation" where both domain specific short forms could fit the context and had very close meanings.

3. *Part-whole / Variation*: For cases like "Support Vector Machines", prediction was "Vector Machines". This is not wrong but doesn't fit the tokens "S, V, and M". The data also had variations like plural form etc. of the same long form, thus reducing data amount and consistency.

7.2. End Phase

Note that the CSVs used to analyse the data and how they were generated are on the Github repository as well.

7.2.1 AE

From the wrong cases and the dataset quality we found that special symbols had a role to play in the scores we got. We accordingly tried models without symbols, without symbols except parentheses, and except "=" sign. This was because a lot of acronyms existed within parentheses and on one side of the "=" sign. We found that these new models didn't do much worse than the original and the below findings confirm how this change could be useful if modelled better:

1. **Better**: A lot of examples that were harmed by special symbols like "?" or *list*ing out symbols like "-", ":", and "," were changed for the better, like: *? Abbreviation features (ABB) : For ... checked whether*, where the original model couldn't factor the word *Abbreviation* in the long form, but on the edited dataset, it could.

2. **Neutral**: It also helped in other cases, but still did not match with the gold data, we think these examples crop up because of the annotation quality: *3 Experimental Results ... the Hong Kong City University (CityU) corpus ... Segmen -*, here our model marked *City University* as the long form, while expected annotation was only *City*.

3. **Worse**: The models without "=" showed how "=" was important, as our model didn't understand the importance of short forms equating to long ones: *Output (0 <x <1) Figure 3 Neural network architecture (* ***DA = descriptor array*** *of 20 items) ..*

We also tried to see if a more generic BERT (as compared to Sci-Bert) could help legal AE task better, but we found out that Sci-Bert was still better on the task. The following examples show how the legal AE benefited from the model, as it was trained on more regular data:

1. The Sci-Bert model understood the intuitive mapping between the letters of an acronym and its corresponding initials of the long form: *Coordinating Committee for Geoscience Programmes in East and Southeast Asia (CCOP) ... Agency (IAEA)*, here the BERT based model skipped the word *Coordinating*.

2. Sci-Bert also identified orphan short forms, without the corresponding long forms. As those are also highly occurring in scientific data: ... *the eTIR international system ... Service) : the eTIR international system will ...*, here the BERT model skipped short tags for the term *eTIR*.

7.2.2 AD

From the analysis in mid phase, we found out that augmenting data would solve a lot of issues and as seen, the AD model after augmentation did very well. Very few mismatches came up:

1. **Legal**: There were mostly no errors, out of the few most very a result of not being grouped by the augmentation Levenshtein process like: *dermal* & *dermal toxicity*. While very few like: *inland transport committee* & *international trace centre*, were actually wrongly identified.

2. **Scientific**: In contrast to legal data, a majority of the mismatches very wrongly identified, like: *hierarchical dirichlet tree* & *hindi dependency treebank*, whereas a few like: *longest common sub* & *longest common subsequence*, very the result of Levenshtein not being able to group them.

To solve these, we feel we need a modified method to group and a cleaner Scientific dataset.

References

[1] Nicholas Egan and John Bohannon. Primer ai's systems for acronym identification and disambiguation, 2021. 1

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 1

[3] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[4] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms, 2019. 1

[5] Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. Bert-based acronym disambiguation with multiple training strategies, 2021. 1

[6] Veyseh, Amir Pouran Ben. SDU '22 Shared Task, 2021. 1

[7] Danqing Zhu, Wangli Lin, Yang Zhang, Qiwei Zhong, Guanxiong Zeng, Weilin Wu, and Jiayu Tang. Atbert: Adversarial training bert for acronym identification winning solution for sdu@aaai-21, 2021. 1