CABACE: Injecting Character Sequence Information and Domain Knowledge for Enhanced Acronym and Long-Form Extraction

Nithish Kannen[1], Divyanshu Sheth[1], Abhranil Chandra[1] and Shubhraneel Pal[1]

[1]*Indian Institute of Technology Kharagpur*

Abstract

Acronyms and long-forms are commonly found in research documents, more so in documents from scientific and legal domains. Many acronyms used in such documents are domain-specific, and are very rarely found in normal text corpora. Owing to this, transformer-based NLP models often detect OOV (Out of Vocabulary) for acronym tokens, especially for non-English languages, and their performance suffers while linking acronyms to their long forms during extraction. Moreover, pre-trained transformer models like BERT are not specialized to handle scientific and legal documents. With these points being the overarching motivation behind this work, we propose a novel framework **CABACE**: **C**haracter-**A**ware **B**ERT for **AC**ronym **E**xtraction, which takes into account character sequences in text, and is adapted to scientific and legal domains by masked language modelling. We use an objective with an augmented loss function, adding max loss and mask loss terms to the standard cross-entropy loss for training CABACE. We further leverage pseudo labelling and adversarial data generation to improve the generalizability of the CABACE framework. Experimental results prove the superiority of the proposed framework in comparison to various baselines. Additionally, we show that the proposed framework is better suited than baseline models for zero-shot generalization to non-English languages, thus reinforcing the effectiveness of our approach. Our team BacKGProp secured the highest scores on the French dataset, second-highest on Danish and Vietnamese, and third-highest in English-Legal dataset on the global leaderboard for the acronym extraction (AE) shared task at SDU AAAI-22.

Keywords

Acronym Extraction, Scientific Documents, Multilingual, Language Modelling, Zero-Shot, Character Embeddings, Adversarial, Information Retrieval

1. Introduction

Acronyms are short forms used to represent a longer sequence of words in documents, for brevity. Most commonly, they are generated by joining the starting letter/letters of each word in their long-form. Such shorthand notations help writers save space and avoid redundant mentions of long-forms in text, especially in scientific papers, which have a page/word limit. A large number of acronyms that occur in scientific and legal documents are domain-specific and are almost never found in common text corpora. As a result, these acronyms often go into the (OOV) – Out of Vocabulary category of NLP models, especially for languages other than English. However, these acronyms quite often form the subject of sentences and play a crucial role in document understanding or text analytics in the scientific domain [1].

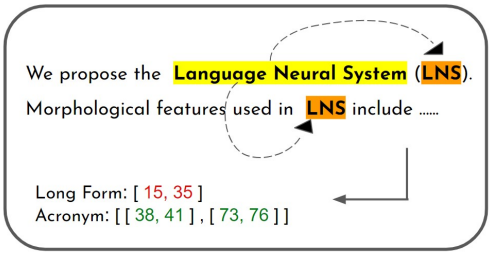

Figure 1: An example sentence depicting long-forms and acronyms appearing together. The goal of the task is to extract the character span indices of these occurrences from documents.

In order to avoid misconceptions among readers, most documents provide the long form of rare acronyms at least at the time of their first mention. As a result, it is important that NLP systems built for scientific document understanding take this into account and look within the documents themselves to understand acronyms and their provenance. For example, in Fig 1, the long-form, Language Neural System is introduced to the reader, and it is referred to by an acronym, LNS thereafter. Systems that can identify acronyms within the passage, and can further link these to their corresponding

Proceedings of SDU@AAAI-22, February 22 - March 1, 2022

nithishkannen@iitkgp.ac.in (N. Kannen);
shethdivyanshu2000@iitkgp.ac.in (D. Sheth);
abhranil.chandra@iitkgp.ac.in (A. Chandra);
shubhraneel@iitkgp.ac.in (S. Pal)

long-forms, if present, would enable models to comprehend scientific documents much better. The ease of availability of scientific documents on arXiv[1] and other open-access archives have led to increased research interest on scientific documents [1, 2, 3].

Although acronym extraction has been studied in the past, the datasets and models used have mostly been limited to the biomedical domain, overlooking challenges in other domains such as science or legal. With this very objective in mind [1] made the largest manually annotated acronym identification and disambiguation dataset publicly available at the SDU Workshop at AAAI-21, which saw systems improving over the baselines. The superiority of human performance in comparison to the best system proposed was highlighted, validating the scope of future research. At SDU-AAAI-22, the acronym extraction task is extended to 6 different languages, namely English, French, Spanish, Danish, Vietnamese and Persian, making it the first publicly available multilingual benchmark for acronym extraction on scientific documents [4] . Given that multilingual NLP/low-resource NLP is starting to pick up pace, such a multilingual benchmark further enables us to test the efficacy of systems on a diverse set of languages.

In this paper, we elucidate our unified framework CABACE: Character-Aware BERT for ACronym Extraction, adapted for 5 languages (all provided in the shared task, except Persian), modelling the acronym extraction (AE) task as a sequence labelling problem. As mentioned earlier, many of the acronyms in scientific text rarely occur in the vocabulary of BERT/Multilingual BERT (mBERT) [5]. Towards this end, we utilise character sequence information of tokens, and aggregate individual character embeddings using a convolutional neural network (CNN) layer, followed by max-pooling. In this way, we inject character sequence information into our model along with mBERT token embeddings. We leverage domain-specific language modelling to enrich mBERT embeddings with domain knowledge, and further improve model generalizability using pseudo labelling and adversarial data augmentation. Additionally, we perform masking of tokens with positive labels (acronyms or long-forms) to encourage CABACE to pay higher attention to the context, and use an objective with an augmented loss function, adding max-loss and mask-loss terms to the standard cross entropy loss. Finally, we also evaluate and provide our system performances on zero-shot transfer from English to the French, Spanish, Danish and Vietnamese datasets. Our proposed model achieved the highest score on the French dataset, and the second highest on Danish and Vietnamese, third highest on Legal English, and fourth on Spanish and Scientific English on the AE Shared Task [6][2] at SDU AAAI-22. We make our code publicly available[3].

To summarize, the key contributions of this work are:

- We propose a novel framework CABACE that leverages mBERT and aggregates character embeddings using CNN and max-pooling to form character-aware token embeddings, which are used along with mBERT token representations. We also inject domain knowledge via masked language modelling performed on scraped data, which is shown to improve AE performance.
- We further use an objective with an augmented loss function, adding max and mask loss terms in addition to standard cross-entropy loss, that results in improved performance when paired with CABACE. We use pseudo labelling and adversarial data generation to improve model generalizability.
- We test the zero-shot generalization efficacy of CABACE across languages and show that it performs better than vanilla mBERT. To the best of our knowledge, this is the first work reporting zero-shot efficacy of acronym extraction systems on a multilingual benchmark.
- We perform extensive experiments on the AE task of SDU-22, achieving state-of-the-art results in both normal and zero-shot settings, demonstrating the effectiveness of our approach.

2. Related Work

Models trained for acronym extraction tasks, like most NLP approaches, can be divided into three major categories: 1) rule-based methods [7, 8], 2) machine learning methods that use text-based features [9, 10], 3) deep learning methods. [11] used BERT and SciBERT with BIO less tagging and blending in an ensemble framework for acronym identification. [12] experimented with multi-task learning, feature engineering and CRF, and found that feature-based methods handled the task well. With the advent

[1] https://arxiv.org/

[2] https://sites.google.com/view/sdu-aaai22

[3] https://github.com/nitkannen/BacKGProp-AAAI-22

Table 1
Key statistics of the released datasets for the Danish, English, French, Spanish and Vietnamese languages, which we evaluate our systems over. The English data has two splits, legal and scientific. We report the average word length and the average number of acronyms and long-forms for a datapoint in each dataset. We also report the number of datapoints containing both acronyms and long-forms, only acronyms, and those with neither acronyms nor long-forms on the train+dev combined set.

Dataset	Train Set Size	Dev Set Size	Test Set Size	Avg. Word Length	Avg. num. of Ac.	Avg. num. of LF	Num. with Ac. + LF	Num. with only Ac.	Num. without Ac./LF
Danish	3082	385	386	64.04	2.04	0.69	2215	915	336
Eng-Leg.	3564	445	446	66.30	2.68	1.48	3913	86	10
Eng-Sci.	3980	497	498	29.56	1.93	1.44	4457	19	1
French	7783	973	973	80.47	2.79	1.74	8585	161	10
Spanish	5928	741	741	85.39	2.18	1.57	6649	16	4
Viet-namese	1274	159	160	33.80	1.05	0.05	215	807	410

of large pretrained language models such as BERT [5], which is a bidirectional multi-layer transformer encoder that achieved state-of-the-art results on a number of benchmarks at the time, most NLP tasks have seen use of transformers-based architectures. [13] used the BERT model with adversarial samples that were created by perturbing existing inputs such that the loss of the model would increase on those samples, to improve the robustness of BERT. Their system was the winning solution to the acronym identification shared task at SDU AAAI-21.

3. Dataset Statistics and Task Description

The organizers of SDU-22 provide acronym extraction (AE) datasets for the shared task in 6 languages, namely English, Spanish, French, Danish, Vietnamese, and Persian There are two distinct splits to the English data, one for texts from the scientific English domain and one for texts from the legal English domain. All datasets are provided in the form of json files, with each individual datapoint having raw text, an ID, and a list of ground truth acronyms and long-forms present in the raw text. Ground truth acronym and long-forms for each datapoint are provided as separate lists, with elements in the list being of the form – [starting character index, ending character index] for each acronym/long-form in the text. The objective of the acronym extraction task is to extract character spans for each identified acronym and long-form in given sentences. Table 1 lists key statistics of the datasets provided in 5 languages.

4. Methodology

Given the input sentence, our objective is to jointly predict the character span of acronyms and long forms present in the text. Towards this goal, we first explain how we formulate the problem. Then we go into the details of the proposed CABACE framework, following which we describe techniques we adopted for robustness and to reduce overfitting, i.e., pseudo-labelling and adversarial data generation.

4.1. Problem Formulation

With the given dataset being annotated with character span indices of acronyms and long-forms, we preprocess the dataset to convert it into sequence tags to model it as a sequence labelling problem. Specifically, we use the BIO tagging scheme [14] for labelling all tokens in the target sequence. We have 5 possible label tags: 0) O-None, 1) B-Acronym, 2) I-Acronym, 3) B-Longform, 4) I-Longform, where 'B' stands for begin, and 'I' stands for inside. The first token for an acronym/long-form would be given a B-label, and the rest would be given I-labels. For all our experiments, we use mBERT that uses the WordPiece tokenizer, whose property we leverage for converting character spans to BIO-tags as explained in Algorithm 1.

The evaluation script provided by the task organizers uses character spans to calculate eval. metrics, so we convert the sequence tags back to character span index after model prediction. For this, we leverage offset map returned by BertTokenizer. More specifically, for each positive prediction by our

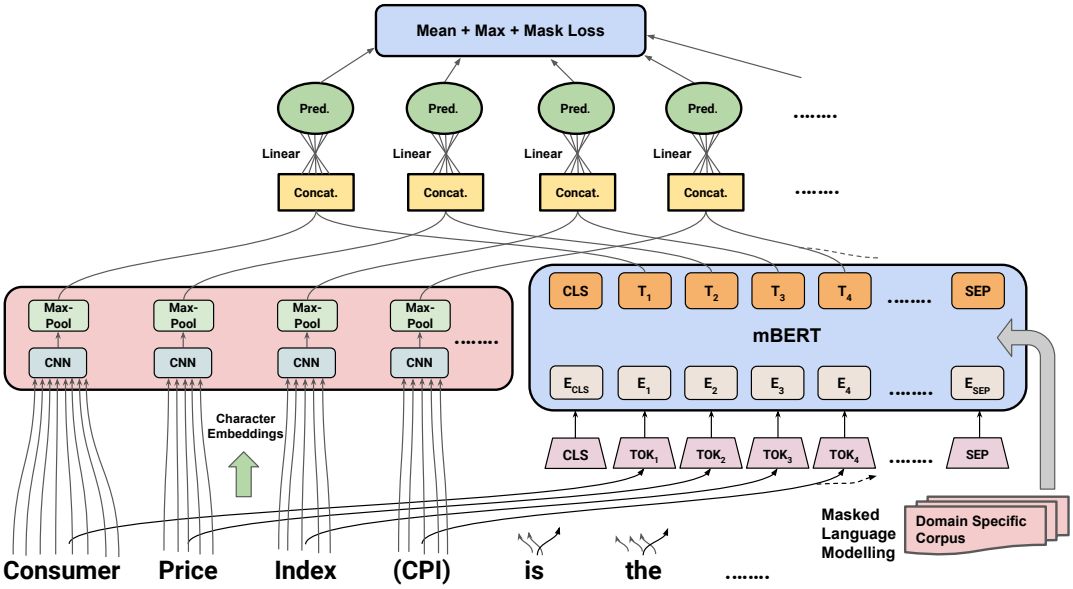

Figure 2: The CABACE Architecture. Input tokens are passed to mBERT (right) and to the CNN & max-pooling layers (left) character-by-character (using character embeddings). The resulting outputs from both are concatenated and passed through a prediction layer (linear + softmax) before computing the augmented loss function. Note that the token '(CPI)' gets split into sub-words by mBERT tokenizer.

Algorithm 1 Conversion of character spans to BIO tags

Input: *Sentence*, *Tokenizer*, *AcronymList*
Output: *target*

```
tokens = Tokenizer(Sentence)
target = [0] * len(tokens)
for each ac_text in Acronym_List do
    ac_tokens = Tokenizer(ac_text)
    for each i in range(len(tokens)) do
        if tokens[i : i + len(ac_tokens)] = ac_tokens then
            FillBIO(i, i + len(ac_tokens), target)
        end if
    end for
end for
return target
```

model (either acronym or long form), we use the token's span to finally compute the character span range of predicted acronyms/long forms.

4.2. The CABACE Framework

We now present CABACE, Character-Aware BERT for ACronym Extraction. Our proposed framework takes as input the tokens of the sentence, and classifies each token into one of the 5 possible labels. For this we augment mBERT with the following components as in Fig. 2: 1) Masked language modelling, to inject domain specific knowledge into mBERT to make it better equipped to handle scientific documents, 2) Character-aware token embeddings, to provide information about character sequences in each token, 3) Augmented loss and masking, motivated by the presence of rarely used notations as acronyms. We describe these components in great detail in the upcoming sections.

4.3. Domain-Specific Language Modelling

To inject scientific domain knowledge into mBERT, which is otherwise not a significant portion of mBERT's pretraining, we perform domain-specific language modeling. For this, we scrape sentences containing acronyms from the web. For scientific English, we scrape data from the arXiv API[4] and for French, Spanish, and Danish, we scrape data from Wikipedia. Then, from the scraped sentences, we filter out those sentences which contain a probable acronym or long-form. A sentence containing a word with more than 50% capitalized letters was used as the criteria to detect sentences with a probable acronym and long-form. We also use the acronym identification data from last year's SDU-21. We merge all these datasets for each language with the

[4] https://arxiv.org/help/api/

datasets provided for this shared task. The total number of sentences thus obtained is 25009 for English Scientific, 4455 for English Legal, 19769 for French, 17411 for Spanish, 15196 for Danish and 1593 for Vietnamese. With this data, we perform masked language modeling (MLM), following the same procedure as described in [5]. We start the training from public mBERT checkpoint and train for 6 epochs. The resulting model weights are saved and are an integral part of the CABACE framework. We use these LM checkpoints with CABACE in all our experiments unless mentioned otherwise.

Each author must be defined separately for accurate metadata identification. Multiple authors may share one affiliation. Authors' names should not be abbreviated; use full first names wherever possible. Include authors' e-mail addresses whenever possible.

4.4. Character-Aware Token Embeddings

A variety of acronyms found in scientific documents often have repeated suffices/prefixes common to them that can be important signals used to detect such acronyms. For example, the acronyms MIT, IIT and NIT have common suffices that can be overlooked if WordPiece tokenizer of mBERT[5] fails to segment out the common suffix 'IT' that serves as an important signal to detect the acronym. Apart from this, characters/sub-tokens that are out of mBERT's vocabulary get mapped to the [UNK] token, where important case-specific or character specific information crucial for the task can be lost. Such cases or more common in low-resource languages like Vietnamese. Motivated by these shortcomings, we propose to additionally inject character-aware token embeddings that are concatenated with mBERT final layer token embeddings before being passed to a prediction layer.

With reference to Fig. 2, we pass the character embeddings of each character of a token to a convolutional neural network (CNN) layer of a fixed filter size. For pre-trained character embeddings, we use the FastText library[6], the dimension of the character embeddings being 300. We pad each input token to a constant character length. Hence, for a token of character length 16 (where max pad length is also 16), we would pass a 16 X 300 dimensional vector $\in \mathbb{R}^{16\times300}$ into the CNN. We then use a max-pooling layer that picks out the CNN filter output that contains the highest signal. This way, we get one embedding vector for each token in the input, which likely contains the most important character n-gram signal from the token (where 'n' refers to the CNN filter size). Let

$$\mathscr{D} = [tok_1, tok_2, ...tok_{k-1}, tok_k]$$

where $\mathscr{D}$ is a sentence with k tokens. The computation for each token tok_i can be summarized by the following equations, where $\tilde{h}_i$ is passed into a linear layer for classification:

$$\tilde{s}_i = BERT(tok_i)$$
$$\tilde{e}_i = MaxPool(CNN(tok_i))$$
$$\tilde{h}_i = \tilde{s}_i \| \tilde{e}_i$$

4.5. Augmented Loss and Masking

Compared to commonly used words, acronyms occur with very low-frequency in the training corpora, and with even lower frequency in the test corpora. As a result, the model must be encouraged to pay higher attention to the context around acronyms to reduce the dependence on the acronym token itself during prediction. Such a feature could be applicable to long forms as well, although with less significance. Towards this end, we randomly mask 10% of tokens with positive labels (B- or I- tokens) during the training phase. This encourages the model to rely less on the token itself, and more on the context leading up to an acronym/long form. Inspired by the successful amendment to the loss function in previous works [15], we additionally append a max term that adds the maximum loss value across all token labels from a particular example to the standard cross entropy loss. This ensures that the model learns more from wrong predictions with high losses, as opposed to an uniformly weighted loss that doesn't special pay attention to the token with the highest loss. Let

$$\mathscr{L} = [Loss(tok_1), Loss(tok_2), ..., Loss(tok_n)]$$

where Loss(.) is given by: $Loss(\hat{p}) = -p\ log(\hat{p})$

Our new objective function contains a weighted addition of max and mask loss terms along with cross entropy.

$$CrossEntropyLoss = mean(\mathscr{L})$$
$$MaxLoss = max(\mathscr{L})$$
$$MaskLoss = \sum_{i=[\text{MASK}]} Loss(tok_i)$$

The max term is weighted by λ_{max} and the mask term is weighted by λ_{mask} which are hyperparameters. The overall augmented loss is given by the following equation.

[5]https://huggingface.co/bert-base-multilingual-cased/blob/main/tokenizer.json

[6]https://fasttext.cc/

$$\begin{aligned} AugmentedLoss &= CrossEntropyLoss \\ &+ \lambda_{max} MaxLoss \\ &+ \lambda_{mask} MaskLoss \end{aligned}$$

4.6. Pseudo-Labelling

Pseudo-labelling [16] is where additional data is created by running a trained model on unseen data and using the model's high-confidence predictions on the unseen data as ground-truths, adding them to the training dataset. We train our models from the public mBERT checkpoint on the training dataset of the original SDU-22 dataset. The trained models are used on the unlabeled scraped data to identify acronyms and long-forms in them. We append the datapoints with high-confidence predictions back to our training set. Pseudo-labeling is a very useful semi-supervised data generation technique [17] that helps particularly when working with low-resource datasets.

4.7. Adversarial Data Generation

We perform adversarial data generation for the English datasets. Adversarial training is a useful technique that enhances the robustness [18] of models by adding adversarial samples to the training data. An adversarial example is an instance with small, intentional feature perturbations that induces the model to make a false prediction [19]. In the procedure of adversarial training, input samples are first mixed with some small perturbations to generate adversarial samples. The model is then trained with both the original input sample and generated adversarial samples [13]. We use the embedding function of the augmenter class from TextAttack [20] to generate text by replacing words with neighbors in the counter-fitted embedding space consisting of GloVe vectors, with a constraint to ensure their cosine similarity is at least 0.8. One such example is:
Original Text: "We conduct a showcase study of dialectal language in online conversational text by investigating African-American English (AAE) on Twitter."
Generated Adversarial Example: "We conduct a case study of dialectal language in online conversational text by scrutinize African-American English (AAE) on Twitter."

5. Experiments

We compare our model with 3 baselines explained in the next section. We then go over the evaluation metrics and implementation details, followed by experiments to test cross-lingual zero-shot efficacy.

5.1. Baselines

We compare our proposed model with 3 different baselines. These are explained in the coming sections:

5.1.1. Rule-Based:

We report the results obtained by the rule-based baseline provided by the organizers of the shared task[7]. This system uses hand-picked rules for extracting acronyms and long-forms. Words that have >60% capitalization are selected as acronyms, and if the initial characters of the preceding words before an acronym can form the acronym, those words are selected as the long-form.

5.1.2. Vanilla mBERT for Sequence Labelling:

Inspired by the success of BERT [5] for sequence labelling in acronym identification shared task of SDU-21 [13], as well as its ability to effectively aggregate contextual information from texts, we consider it as a baseline in our experiments. We run our inputs through mBERT, and use mBERT's final layer tokens, which are then passed to a linear classifier followed by softmax to generate 1 out of the 5 BIO tags as prediction for that token. Problem formulation and pre-processing is the same as in CABACE. We use Multilingual-BERT (bert-base-multilingual-cased)[8] and refer to this model in our experiments as Vanilla mBERT.

5.1.3. Seq-to-Seq:

Previous works have reported sequence to sequence approach using the encoder-decoder architecture as an alternative for the sequence tagging scheme [21, 22]. Following these works, we use a transformer-based generative framework to autoregressively decode the acronyms and long-forms present in input sentence. For the example given in Fig.1, the ground truth target sequence would follow the template string: "<Acronyms> LNS <Long-Forms> Language Neural System", where "<Acronyms>" and "<Long-Forms>" contain the predicted acronyms and long-forms from the sentence, separated by a comma. We decode the output by searching for occurrences of the predicted

[7] https://github.com/amirveyseh/AAAI-22-SDU-shared-task-1-AE/blob/main/code/baseline.py

[8] https://huggingface.co/bert-base-multilingual-cased

Table 2
Hyperparameters used for the CABACE model

Hyperparameter	Value
Batch size	8
Token character len.	16
CNN filter size	4
λ_{max}	2.0
λ_{mask}	1.0
Mask rate	0.1
Learning rate	2e-5

acronyms and long-forms and detecting their character spans in the input text. We use mT5 for our experiments [23].

5.2. Zero-Shot Transfer to Non-English Languages

Multilingual models like mBERT have a shared embedding space across languages, which they leverage to learn language-agnostic properties of the task along with language-specific features. Zero-shot transfer is a useful way to test how well multilingual models generalize to unseen languages [24]. To this end, we conduct experiments by training models on a combined English-Legal + English-Scientific dataset, and testing their performances on the other languages, i.e., Danish, French, Spanish and Vietnamese datasets. We comparatively evaluate Vanilla mBERT and CABACE this way to test their zero-shot efficacy.

5.3. Evaluation Metrics

The metrics reported here is identical to the ones provided by the organizers of the shared task[9]. The metrics used were precision, recall and F1-score for the acronyms and long-forms, both individually and combined together. The script uses exact match to pair predicted spans with the gold spans.

5.4. Implementation Details

All experiments were done using Pytorch 1.10.0+cu11.1 on Google Colab GPUs (NVIDIA Tesla P100-PCIe-16GB). For the mBERT and mT5 checkpoints, we used the HuggingFace transformers 4.12.5 library[10]. We used the base-multilingual-cased version for mBERT and the base version for mT5. HuggingFace datasets 1.15.1 was used to handle dataset processing. For fine-tuning, we used the AdamW optimizer with learning rate 2e-5 along with a learning rate scheduler with warmup, wherein the learning rate decreases from its initial value to 0 after a warmup period of 0 to the initial value. We set gradient clipping to 1.0 to prevent exploding gradients. The maximum sequence length for finetuning mBERT was taken as 512. The maximum sequence length for both input and output in mT5 was taken to be 600. For masked language modelling, we used the same scheme as mentioned in Devlin et al. [5]. Table 2 lists the hyperparameters used in our experiments. All results are reported on the development set of the shared task.

[9]https://github.com/amirveyseh/AAAI-22-SDU-shared-task-1-AE/blob/main/code/scorer.py
[10]https://huggingface.co/transformers/

6. Results and Discussion

Table 3 lists comparative results between CABACE and the Rule-based model, the Seq-to-seq (mT5) and Vanilla mBERT baselines. We find that CABACE improves significantly upon the three baselines in all 6 datasets we evaluate the models

Table 3
Comparison of CABACE with the three baseline models. Reported scores are on the dev set. CABACE is seen to outperform all baseline methods in all datasets.

Datasets	Model	Precision	Recall	F1
Danish	Rule-Based	0.1000	0.0600	0.0800
	Seq-to-Seq (mT5)	0.5773	0.6821	0.6254
	Vanilla mBERT	0.9285	0.9515	0.9398
	CABACE (Ours)	**0.9435**	**0.9572**	**0.9503**
English Legal	Rule-Based	0.3200	0.1000	0.1600
	Seq-to-Seq (mT5)	0.7024	0.6398	0.6697
	Vanilla mBERT	0.8592	0.8727	0.8659
	CABACE (Ours)	**0.8593**	**0.8756**	**0.8681**
English Scientific	Rule-Based	0.3300	0.1500	0.2000
	Seq-to-Seq (mT5)	0.8000	0.7455	0.7718
	Vanilla mBERT	0.8108	0.8535	0.8316
	CABACE (Ours)	**0.8282**	**0.8876**	**0.8509**
French	Rule-Based	0.2200	0.0600	0.1000
	Seq-to-Seq (mT5)	0.7891	0.6771	0.7288
	Vanilla mBERT	0.9133	0.9168	0.9150
	CABACE (Ours)	**0.9387**	**0.9423**	**0.9405**
Spanish	Rule-Based	0.1700	0.0700	0.1000
	Seq-to-Seq(mT5)	0.7544	0.6604	0.7043
	Vanilla mBERT	0.8656	0.8770	0.8712
	CABACE (Ours)	**0.8842**	**0.9029**	**0.8934**
Vietnamese	Rule-Based	0.8200	0.3900	0.5300
	Seq-to-Seq(mT5)	0.50	-	-
	Vanilla mBERT	0.7852	0.6589	0.7165
	CABACE (Ours)	**0.9077**	**0.7839**	**0.8413**

Table 4
Fine-grained performance metrics using CABACE Combined depicts the overall score, while Acronyms and Long-Forms extraction metrics are individually reported.

Datasets	Fine-grained	Precision	Recall	F1
Danish	Acronyms	0.9637	0.9809	0.9722
	Long-Forms	0.9234	0.9336	0.9284
	Combined	0.9435	0.9572	0.9503
English Legal	Acronyms	0.8870	0.9126	0.8996
	Long-Forms	0.8190	0.8386	0.8287
	Combined	0.8593	0.8756	0.8681
English Scientific	Acronyms	0.9038	0.9299	0.9167
	Long-Forms	0.7968	0.8222	0.8093
	Combined	0.8282	0.8876	0.8509
French	Acronyms	0.9599	0.9491	0.9541
	Long-Forms	0.9208	0.9269	0.9173
	Combined	0.9387	0.9423	0.9405
Spanish	Acronyms	0.9308	0.9447	0.9377
	Long-Forms	0.8376	0.8610	0.8491
	Combined	0.8842	0.9029	0.8934
Vietnam-ese	Acronyms	0.9821	0.9429	0.9621
	Long-Forms	0.8333	0.6250	0.7143
	Combined	0.9077	0.7839	0.8413

upon. Note that the improvement with CABACE in a language like Vietnamese is much more than in English-Legal. This can be attributed to the fact that mBERT classifies many of the tokens in Vietnamese into [UNK], consequently losing crucial information. Injecting character embeddings for these tokens enables CABACE to perform much better than Vanilla mBERT. Vanilla mBERT performs better than the Seq-to-seq model, which in turn ups the rule-based model's performance, in all datasets.

Table 4 shows fine-grained results of the CABACE architecture on the 6 datasets – precision, recall and F1-scores for acronyms and long-forms are provided separately. We observe that CABACE is able to perform very well on extracting acronyms, but loses some points while detecting long-forms, which could be due to lack of components in the architecture that are specialized to focus on the acronym-long form interactions during extraction. Recall scores are seen to be always higher than precision scores, except on the Vietnamese dataset.

A comparison of Vanilla mBERT vs CABACE for zero-shot performance can be found in Figure 3. We find that CABACE outperforms Vanilla mBERT in 3 out of 4 languages for cross-lingual zero-shot transfer. Surprisingly, the zero-shot performance in Spanish using both models isn't much less than performances of the models when trained on the Spanish dataset. However, languages like Vietnamese and Danish see significant zero-shot performance

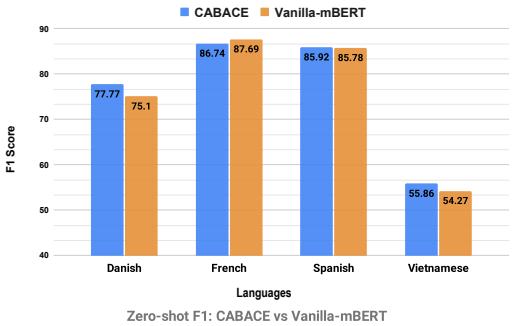

Figure 3: This figure depicts the zero-shot performance of CABACE as compared to Vanilla mBERT on the non-English languages, with training being done on a combined English-Legal + English-Scientific dataset.

Table 5
CABACE ablation study on French

Ablations	Precision	Recall	F1
Ours	0.9387	0.9423	0.9405
Ours w/o max loss	0.9379	0.9426	0.9402
Ours w/o mask loss	0.9350	0.9403	0.9376
Ours w/o Char Embd	0.9361	0.9417	0.9389
Ours w/o LM	0.9373	0.9432	0.9402

Table 6
CABACE ablation study on English-Scientific

Ablations	Precision	Recall	F1
Ours	0.8282	0.8876	0.8509
Ours w/o max loss	0.8223	0.8811	0.8450
Ours w/o mask loss	0.8330	0.8822	0.8503
Ours w/o Char Embd	0.8254	0.8729	0.8485
Ours w/o LM	0.8073	0.8658	0.8355

drops compared to in-dataset performance. Possible reasons could be due to lexicons and grammar in these languages being relatively distant to those in English, and relative similarity of Spanish and English.

6.1. Ablation Study on CABACE

We perform an ablation study to better interpret how individual components contribute to performance improvements. The steps we perform include comparing our full model (CABACE), CABACE without the max loss component, CABACE without

the mask loss component, CABACE without the character embeddings, and CABACE without language modelling. We perform the ablation study on the French dataset (where we top the leaderboard) and on the English-Scientific dataset. Both these studies show similar trends. As seen in tables 5 and 6, removing augmented loss i.e. max and mask loss causes a significant drop in the scores, and so does removing the character embeddings. Not including domain-specific language model checkpoints leads to drops in scores too, as expected.

7. Conclusion and Future Work

In this paper, we introduce the CABACE framework for acronym and long-form extraction that integrates character-level information with mBERT representations and uses domain-specific language modelling and an augmented loss function, with pseudo labelling and adversarial data generation for improved generalizability. Experimental results establish the supremacy of our framework over several baselines on 6 datasets spanning 5 languages. We also evaluate zero-shot cross-lingual efficacy of our proposed model and find that it outperforms baseline mBERT results in 3 out of 4 cases. Our system merits the top spot in French, second place in Danish and Vietnamese and third place in Legal English leaderboards on the AE shared task at SDU AAAI-22.

References

[1] A. P. B. Veyseh, F. Dernoncourt, T. H. Nguyen, W. Chang, L. A. Celi, Acronym identification and disambiguation shared tasks for scientific document understanding, 2021. arXiv:2012.11760.

[2] H. Timmapathini, A. Nayak, S. Mandadi, S. Sangada, V. Kesri, K. Ponnalagu, V. G. Venkoparao, Probing the spanbert architecture to interpret scientific domain adaptation challenges for coreference resolution, in: SDU@AAAI, 2021.

[3] X. Li, M. Daoutis, Unsupervised keyphrase extraction and clustering for classification scheme in scientific publications, 2021. arXiv:2101.09990.

[4] A. P. B. Veyseh, N. Meister, S. Yoon, R. Jain, F. Dernoncourt, T. H. Nguyen, MACRONYM: A Large-Scale Dataset for Multilingual and Multi-Domain Acronym Extraction, in: arXiv, 2022.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[6] A. P. B. Veyseh, N. Meister, S. Yoon, R. Jain, F. Dernoncourt, T. H. Nguyen, Multilingual Acronym Extraction and Disambiguation Shared Tasks at SDU 2022, in: Proceedings of SDU@AAAI-22, 2022.

[7] K. Taghva, J. Gilbreth, Recognizing acronyms and their definitions, International Journal on Document Analysis and Recognition 1 (1999) 191–198. URL: https://doi.org/10.1007/s100320050018. doi:10.1007/s100320050018.

[8] A. S. Schwartz, M. A. Hearst, A simple algorithm for identifying abbreviation definitions in biomedical text, 2003.

[9] C.-J. Kuo, M. H. Ling, K.-T. Lin, C.-N. Hsu, Bioadi: a machine learning approach to identifying abbreviations and definitions in biological literature, BMC Bioinformatics 10 (2009) S7. URL: https://doi.org/10.1186/1471-2105-10-S15-S7. doi:10.1186/1471-2105-10-S15-S7.

[10] C. G. Harris, P. Srinivasan, My word! machine versus human versus computation methods for identifying and resolving acronyms, 2019.

[11] A. Singh, P. Kumar, Scidr at sdu-2020: Ideas – identifying and disambiguating everyday acronyms for scientific domain, 2021. arXiv:2102.08818.

[12] F. Li, Z. Mai, W. Zou, W. Ou, X. Qin, Y. Lin, W. Zhang, Systems at sdu-2021 task 1: Transformers for sentence level sequence label, in: SDU@AAAI, 2021.

[13] D. Zhu, W. Lin, Y. Zhang, Q. Zhong, G. Zeng, W. Wu, J. Tang, At-bert: Adversarial training bert for acronym identification winning solution for sdu@aaai-21, 2021. arXiv:2101.03700.

[14] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 260–270. URL: https://aclanthology.org/N16-1030. doi:10.18653/v1/N16-1030.

[15] Q. Wu, Z. Lin, G. Wang, H. Chen, B. F. Karlsson, B. Huang, C.-Y. Lin, Enhanced meta-learning for cross-lingual named entity recognition with minimal resources, 2020. arXiv:1911.06161.

[16] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, 2019. arXiv:1904.04717.

[17] P. Cascante-Bonilla, F. Tan, Y. Qi, V. Ordonez, Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning, 2020. arXiv:2001.06001.

[18] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2015. arXiv:1412.6572.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2014. arXiv:1312.6199.

[20] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, Y. Qi, Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 119–126.

[21] S. Almasian, D. Aumiller, M. Gertz, Bert got a date: Introducing transformers to temporal tagging, 2021. arXiv:2109.14927.

[22] H. Yan, J. Dai, T. ji, X. Qiu, Z. Zhang, A unified generative framework for aspect-based sentiment analysis, 2021. arXiv:2106.04300.

[23] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, 2021. arXiv:2010.11934.

[24] H. Choi, J. Kim, S. Joe, S. Min, Y. Gwon, Analyzing zero-shot cross-lingual transfer in supervised nlp tasks, 2021. arXiv:2101.10649.