GLADIS: A General and Large Acronym Disambiguation Benchmark

Lihu Chen[1], **Gaël Varoquaux**[2], **Fabian M. Suchanek**[1]

[1] LTCI & Télécom Paris & Institut Polytechnique de Paris, France
[2] Soda, Inria Saclay, Université Paris-Saclay, France
{lihu.chen, fabian.suchanek}@telecom-paris.fr
{gael.varoquaux}@inria.fr

Abstract

Acronym Disambiguation (AD) is crucial for natural language understanding on various sources, including biomedical reports, scientific papers, and search engine queries. However, existing acronym disambiguation benchmarks and tools are limited to specific domains, and the size of prior benchmarks is rather small. To accelerate the research on acronym disambiguation, we construct a new benchmark named GLADIS with three components: (1) a much larger acronym dictionary with 1.5M acronyms and 6.4M long forms; (2) a pre-training corpus with 160 million sentences; (3) three datasets that cover the general, scientific, and biomedical domains. We then pre-train a language model, *AcroBERT*, on our constructed corpus for general acronym disambiguation, and show the challenges and values of our new benchmark.

1 Introduction

An acronym is an abbreviation formed from the initial letters of a longer name. For instance, the following two sentences contain the acronym *"AI"*: (1) *This is the product's first true **AI** version, and it understands your voice instantly.* (2) *In the United States, the **AI** for potassium for adults is 4.7 grams.* The long forms (or expanded forms) for the same acronym are *"Artificial Intelligence"* and *"Adequate Intake"*, respectively.

Acronym Disambiguation (AD) is the task of mapping a given acronym in a given sentence to the intended long form. Acronym disambiguation is crucial for downstream tasks such as information extraction, machine translation, and query analysis in search engines (Jain et al., 2007; Islamaj Dogan et al., 2009). Acronym disambiguation is also important for humans: acronyms may make a text more difficult to understand for readers who are not familiar with the specific domain. A study on a Microsoft question answering forum found that

ID	Long Form	Popularity	Domain
1	*Artificial Intelligence*	★★★★★	Computer Science
2	*Adequate Intake*	★★★★	Food and Nutrition
3	*Aromatase Inhibitor*	★★★	Chemistry
4	*Apoptotic Index*	★★★	Biomedicine
5	*Asynchronous Irregular*	★★★	Neuroscience
6	*Amnesty International*	★★	Organization
7	*Anterior Insula*	★★	Biomedicine
8	*Air India*	★★	Organization
9	*Article Influence*	★★	Science
......			
2243	*Agricultural Implement*	★	Agriculture

Table 1: Long form candidates for the acronym *"AI"* from our acronym dictionary. The SciAD benchmark (Veyseh et al., 2020) only includes two long terms (black) in the scientific domain. The popularity is the occurrence frequency in our collected corpora.

only 7% of the acronyms co-occur with their corresponding long forms, which confuses the readers about the meaning of a text (Li et al., 2018).

Acronym Disambiguation has received more attention in the past few years. The first step in acronym disambiguation is usually the creation of a dictionary, i.e., a mapping of each acronym to one or more long forms. Early systems extracted acronyms and their definitions automatically from texts by rule-based (Schwartz and Hearst, 2002) or supervised (Nadeau and Turney, 2005) methods. Once a dictionary is available, acronym disambiguation methods expand acronyms in a given text by capturing the contexts for specific domains, e.g., the enterprise domain (Li et al., 2018), biomedical texts (Jin et al., 2019), and scientific papers (Charbonnier and Wartena, 2018). Madog (Veyseh et al., 2021) was the first general and web-based system, recognizing and expanding acronyms across multiple domains. Several benchmarks have also been constructed, including for the biomedical area (Suominen et al., 2013) and the scientific area (SciAD, Veyseh et al., 2020). Several methods fine-tuned SciBERT (Beltagy et al., 2019) on SciAD to disambiguate acronyms in scientific documents (Pan et al., 2021; Zhong et al., 2021; Li et al., 2021).

Although these works have significantly advanced the progress of acronym disambiguation, they suffer from three main limitations. First, most existing dictionaries (and benchmarks) focus on one specific domain. In real-world applications, however, the input text may be general, cross-domain, or of an unspecified domain (as in search engine queries). Second, existing dictionaries are limited in size. For example, there are only two long forms for the acronym *"AI"* in SciAD (Table 1), which is constructed from arXiv. However, we find that the two long forms *"Asynchronous Irregular"* and *"Anterior Insula"* also appear in scientific papers on arXiv (Girardi-Schappo et al., 2019; Vadovičová, 2014), and the acronym *"AI"* also appears separately without the long form in sentences. In our work, we actually find at least 2 243 different long forms for *"AI"*. Besides, SciAD suffers from the problem of data leakage, because the train and test sets have overlapping pairs of acronym and long form. Finally, current general AD systems such as MadDog (Veyseh et al., 2021) rely on static word embeddings and LSTMs (Long Short Term Memory (Hochreiter and Schmidhuber, 1997)). Thus, they do not leverage pre-training on large corpora, which drives the current state of the art in most NLP tasks with contextual embeddings like BERT (Devlin et al., 2018).

With this work, we aim to improve Acronym Disambiguation along two dimensions: First, we automatically construct GLADIS, a **G**eneral and **L**arge **A**cronym **DIS**ambiguation benchmark that includes a larger dictionary, a pre-training corpus and three datasets covering the general, biomedical, and scientific domains. Our dictionary contains 1.5M acronyms and 6.4M long forms, which trumps existing dictionaries by a factor of 3. We complement this dictionary by three domain-specific datasets for acronym disambiguation, which are adapted from three existing human-annotated and crowd-sourced datasets (Mohan and Li, 2018; Onoe and Durrett, 2020; Veyseh et al., 2020). The pre-training corpus has 160 million sentences with acronyms, collected from the Pile dataset (Gao et al., 2020) with a rule-based algorithm (Schwartz and Hearst, 2002). Second, we propose AcroBERT, the first pre-trained language model for general acronym disambiguation. Our experiments show that this model outperforms existing systems across multiple domains. Our code and data are available at https://github.com/tigerchen52/GLADIS.

2 Related Work

2.1 Acronym Identification and Disambiguation

To expand acronyms, there are usually two subtasks: Acronym Identification (AI), which creates a dictionary of acronyms and their definitions from a given document, and Acronym Disambiguation (AD), which aims to link acronyms in the input text to the correct long forms from a dictionary.

The study of acronym identification has a long history. Early work observed that acronyms and their long forms appear frequently together in a document, as in *"Artificial Intelligence (AI)"*. Based on this pattern, many approaches identify and extract acronyms by using rules (Yeates et al., 2000; Larkey et al., 2000; Pustejovsky et al., 2001; Park and Byrd, 2001; Yu et al., 2002; Schwartz and Hearst, 2002; Adar, 2004; Ao and Takagi, 2005; Okazaki and Ananiadou, 2006; Sohn et al., 2008; Veyseh et al., 2021) or supervised methods (Chang et al., 2002; Nadeau and Turney, 2005; Kuo et al., 2009; Movshovitz-Attias and Cohen, 2012; Liu et al., 2017; Wu et al., 2017; Zhu et al., 2021). In our work, we build on previous work (Schwartz and Hearst, 2002) for Acronym Identification, and focus mainly on disambiguation.

As for acronym disambiguation, early solutions manually designed features to score each pair of acronyms and long forms, by either unsupervised (Jain et al., 2007; Henriksson et al., 2014) or supervised machine learning (Pakhomov et al., 2005; Yu et al., 2007; Stevenson et al., 2009; Finley et al., 2016; Li et al., 2018). Later, deep learning approaches were introduced to the task, using embeddings to represent word sequences. The methods can be categorized as static embedding-based (Wu et al., 2015; Li et al., 2015; Charbonnier and Wartena, 2018) and dynamic embedding-based (Jin et al., 2019; Pan et al., 2021; Zhong et al., 2021; Li et al., 2021), where the former generates fixed representations for words in a pre-defined vocabulary and the latter can represent arbitrary words dynamically based on specific contexts. One main limitation of these methods is that they are domain-specific systems that can be applied only to a certain field such as the biomedical domain or scientific documents. To generalize the system, Ciosici and Assent (2018) presented the Abbreviation Expander, and Veyseh et al. (2021) proposed MadDog, both of which can be used in multiple domains. In this paper, we improve over the performance of

these systems by adapting transformer-based methods and pre-training strategies.

2.2 Existing benchmarks

Most current public datasets for acronym expansion are focused on a particular domain, such as the biomedical domain (Suominen et al., 2013; Wen et al., 2020) or science (Charbonnier and Wartena, 2018; Veyseh et al., 2020). Some works adopt two domain-specific datasets for better evaluations (Ciosici et al., 2019; Veyseh et al., 2022). The main limitation of these benchmarks is two-fold: first, their acronym dictionaries are rather small. For instance, the average number of candidates per acronym in the SciAD benchmark (Veyseh et al., 2020) is 3.15 while in our benchmark the number is greater than 200. Second, there are no AD evaluation sets that cover multiple domains. We also note that, in SciAD, the train and test sets have overlapping pairs of acronym and long form. For example, the pair ⟨CT, Computed Tomography⟩ appears in the training, validation, and test sets.

3 Constructing GLADIS

Our GLADIS benchmark consists of three components: a dictionary, a pre-training corpus, and three domain-specific datasets.

3.1 Dictionary and Pre-training Corpus

We propose an acronym dictionary that addresses the shortcomings of existing dictionaries (Section 2.2) by being (1) cross-domain and (2) large in size. To construct this dictionary, we apply rule-based extraction on a large set of corpora that contain acronym definitions. In this process, we can also obtain a large number of sentences containing acronyms as the pre-training corpus.

Input Corpora. For the textual data source, we use the Pile dataset (Gao et al., 2020), an 825 GiB English corpus constructed from 22 diverse high-quality subsets (see the details of Pile in Appendix A.1). We also make use of structured knowledge from knowledge bases, namely the Alias Table from Wikidata and the Concept Names from UMLS. Both of them contain alternate names for canonical entities, and these may be acronyms or not. To consider only the acronyms, we produce pairs of the canonical name and an alternate name in the form *"canonical form (alternate name)"*. The rule-based algorithm will then decide whether to extract an acronym or not. Table 2 shows the

Subset	Domain	Size (GiB)
Pile-CC	Web Archive files	227.12
Books3	Books	100.96
Github	Open-source codes	95.16
PubMed Central	Biomedical articles	90.27
OpenWebText2	Reddit submissions	62.77
ArXiv	Research papers	56.21
FreeLaw	Legal proceedings	51.15
Stack Exchange	Question-answer texts	32.20
USPTO Backgrounds	Patents	22.90
PubMed Abstracts	Biomedical abstracts	19.26
OpenSubtitles	Subtitles	12.98
Gutenberg (PG-19)	Western literatures	10.88
DM Mathematics	mathematical problems	7.75
Wikipedia (en)	Wikipedia pages	6.38
BookCorpus2	Books	6.30
Ubuntu IRC	Chatlog data	5.52
EuroParl	Proceedings	4.59
HackerNews	Comments of social news	3.90
YoutubeSubtitles	YouTube subtitles	3.73
PhilPapers	Philosophy publications	2.38
NIH ExPorter	Awarded applications	1.89
Enron Emails	Emails	0.88
Wikidata Alias	Alias Table	11.00
UMLS Concept	Biomedical Vocabulary	1.96
Total	-	838.14

Table 2: Sources for acronym extraction. All corpora except Wikidata Alias and UMLS Concept are from Pile (Gao et al., 2020).

statistics of our sources. They cover a wide range of domains including Web pages, books, scientific and biomedical papers, legal documents, etc.

Acronym Extraction. To extract acronyms from the textual sources, we use the rule-based algorithm proposed by Schwartz and Hearst (2002). It assumes that acronyms follow a predictable pattern, e.g., *long form (acronym)* or *acronym (long form)*, and then uses rules to extract candidate pairs by identifying parentheses and surrounding tokens. Experimental results show that this simple algorithm achieves 95% precision and 82% recall, averaged over two datasets. As the method has good results at low time complexity, we decided to not adopt more sophisticated methods. Some extracted samples are shown in Table A2 in the appendix. A manual evaluation on a random sample of 100 extracted sentences yields a precision of 94%.

Dictionary Construction. Next, we build a large-scale acronym dictionary with frequencies (popularity) by merging the extracted outputs. This merger may regroup duplicate long forms for an acronym, e.g., *"convolutional neural network"*, *"convolutional-neural network"* or *"convolutional neural networks"*. Therefore, we merge long forms that are identical after stemming and removing punctuation. In our case, the above three forms are merged into *"convolutional neural network"*. We keep the most frequent, unprocessed, long
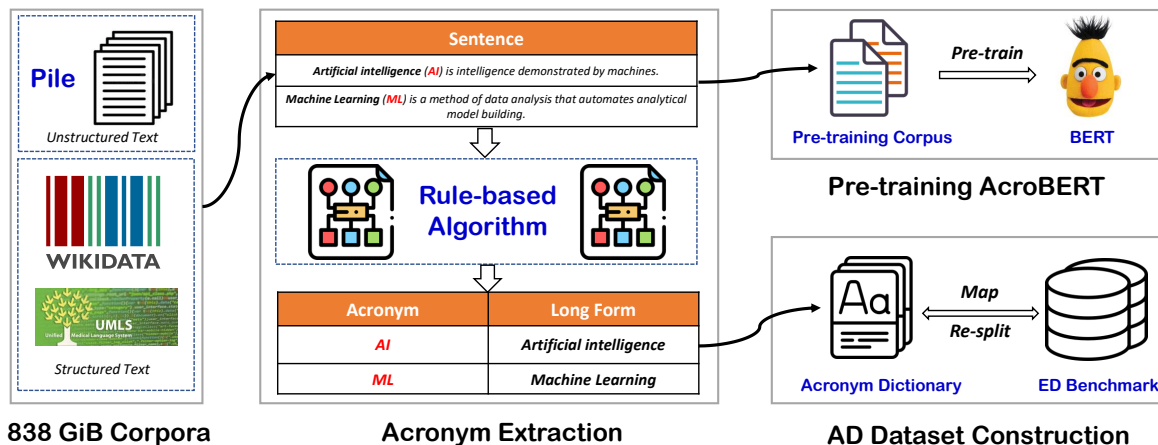
Figure 1: Framework of our benchmark construction. The *"ED"* in the lower right corner means *"Entity Disambiguation"*.

	train	valid	test	unique short form	long forms per acronym	overshadowed ratio
General	13,269	7,024	7,125	1,147	248	29.8%
Scientific	28,023	14,134	14,066	2,922	262	68.7%
Biomedical	6,295	3,150	3,149	2,909	278	27.4%

Table 3: Statistics of our new Acronym Disambiguation Benchmark. The last column shows the ratio of overshadowed samples in the dataset: long forms with the same acronym but not the most popular one.

	Short Form	Long Form	Avg
SciAD (2020)	732	2,308	3.15
MadDog (2021)	426,389	3,781,739	8.87
Ours	1,542,819	6,381,257	4.14

Table 4: Statistics for three acronym dictionaries. The "Avg" column shows the average number of long forms per acronym.

form as the canonical name in our dictionary, discarding other forms. There are still some noisy long forms that cannot be merged, caused by typos and nested acronyms (see Section 7). However, a manual evaluation on a sample shows that 94% of the long forms are clean. If the long forms are weighted by their frequency, the percentage of clean forms increases to 97%. Most notably, almost frequent long forms for a given acronym were clean in our sample. The statistics of our dataset are shown in Table 4. Our resource will be the largest publicly available dictionary for acronyms that covers various domains.

Pre-training Corpus. While building the dictionary, we can also collect the sentences that contain acronyms for pre-training. For example, the following sentence contains the acronym *ELEC*: *"Christie, some legislators and the state*

Election Law Enforcement Commission (ELEC), have joined the comptroller in voicing support for the elimination of the loophole." For pre-training, the long form *Election Law Enforcement Commission* is removed, and we then force the model to restore the long form from our constructed dictionary, based on the input sentence and the acronym. In total, we collect a pre-training corpus with ~160 million sentences. More examples are shown in Table A2.

3.2 Acronym Disambiguation Dataset

We use our acronym dictionary to construct new, larger datasets for evaluating AD systems. To automatically construct the datasets, we adapt the existing two Entity Disambiguation datasets by replacing the long form of entity with the acronym. For example, one sentence in Medmentions (Mohan and Li, 2018) contains the long form of Cerebral Blood Flow: *"The reconstructed volume was then compared with corresponding magnetic resonance images demonstrating that the volume of reduced **Cerebral Blood Flow** agrees with the infarct zone at twenty-four hours"*. The dataset provides the unique ID of this long form in UMLS (*C1623258*), and we use it to find the acronym *CBF* in UMLS.
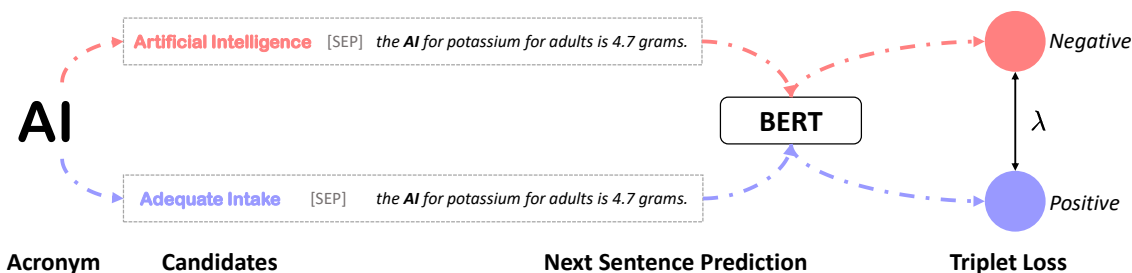
Figure 2: The pre-training strategy of AcroBERT. λ is a margin between positive and negative pairs, here ⟨*Adequate Intake, AI*⟩ and ⟨*Artificial Intelligence, AI*⟩.

Therefore, a new sample can be obtained by replacing the long form with its corresponding acronym.

Specifically, we use the following human-annotated and crowd-sourced datasets:

WikilinksNED Unseen Mentions (Onoe and Durrett, 2020) is an Entity Disambiguation dataset, i.e., a set of text documents that have mentions of entities, together with a reference knowledge base (KB) that contains, for each entity, one or several names. WikilinksNED Unseen Mentions re-splits the WikilinksNED dataset (Eshel et al., 2017) to ensure that all mentions in the validation and test sets do not appear in the training set. This is a large-scale, crowd-sourced ED dataset from websites in various fields, which is significantly noisier and more challenging than prior datasets. The reference KB is Wikidata (or Wikipedia), and we adapt this WikilinksNED Unseen Mentions to an AD dataset in the **general** domain.

Medmentions (Mohan and Li, 2018) is an entity disambiguation dataset of 4,392 PubMed papers that were annotated by professional and experienced annotators in the biomedical domain. The reference KB is UMLS (Bodenreider, 2004), and this is a **biomedical** dataset.

SciAD (Veyseh et al., 2020) is the previously mentioned acronym disambiguation dataset in the **scientific** domain.

SciAD is already an AD dataset, and we only re-split it to avoid data leakage. As for the two ED datasets, they both provide a unique ID to the reference KB for each long form. We then replace the long forms with the acronyms from their corresponding reference KB, i.e., Wikidata and UMLS. To make sure this replacement is correct, we apply the rule-based algorithm (Schwartz and Hearst, 2002) to the pair of long form and acronym again for verification. We manually checked 100 random sentences constructed in this way and did not find problematic cases. Hence, this semi-synthetic construction results in a dataset of natural text in which the long form and the acronym are mutually replaceable in the context. Besides, the pair is added to our dictionary with a frequency of 1 if it does not appear in our dictionary. For the WikilinksNED dataset, we use the taxonomy of YAGO 4 (Pellissier Tanon et al., 2020) to label each long form with a top-level class. For example, *"rhythm and blues"* is a CreativeWork and *"United States Navy"* is an Organization.

We then partition the three datasets separately into training, test, and validation set, ensuring that the acronyms in the training set do not appear in the validation and test sets. We repartition the datasets at the ratio of 6:2:2. Table 3 gives the statistics of this new benchmark. It is not only larger but also more challenging than existing benchmarks, because acronyms in our benchmark have more than 200 candidates on average. Moreover, it contains many *overshadowed forms* (Provatorova et al., 2021), which means that an acronym has to be disambiguated to a long form that is not the most popular long form for that acronym. For example, *"Adequate Intake"* is overshadowed by the more popular form *"Artificial Intelligence"* for the acronym *"AI"*.

4   AcroBERT

We can now capitalize on our dictionary and pre-training corpus to propose a new method for acronym disambiguation. It takes as input (1) a dictionary of acronyms with their long form(s), and (2) a large-scale corpus that contains acronyms (we assume that the boundaries of the acronym have already been recognized). Our goal is to pre-train a language model for acronym disambiguation, which has a strong generalizability across multiple domains.

Model	General				Scientific				Biomedical				Avg	
	Dev		Test		Dev		Test		Dev		Test			
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
BM25 (1995)	29.9	32.6	35.5	25.8	14.1	5.4	17.1	10.7	13.1	8.3	17.0	14.3	21.1	16.2
FastText (2017)	11.3	12.9	18.7	12.7	3.3	0.9	5.7	2.5	0.2	0.1	1.3	0.7	6.8	5.0
MadDog (2021)	28.1	11.7	29.9	23.1	17.8	15.5	22.4	17.9	33.8	19.3	41.2	35.9	28.9	20.6
BERT (2018)	32.3	32.5	37.7	28.2	15.1	5.8	17.6	9.3	3.1	1.3	3.5	2.1	18.2	13.2
Popularity-Ours	35.2	39.1	39.0	43.2	5.5	22.9	4.9	12.3	46.0	61.3	49.9	54.0	30.1	38.8
AcroBERT	**74.7**	**78.8**	**70.0**	**72.0**	**26.9**	**36.6**	**28.8**	**27.4**	**58.4**	**66.0**	**59.9**	**61.4**	**53.1**	**57.0**

Table 5: Performances of the unsupervised setting across different models, measured by macro F1 and Accuracy.

The Pre-training Strategy of BERT. We adapt the BERT model for our purpose. BERT is pre-trained by using two unsupervised tasks, Masked Language Model (MLM) and Next Sentence Prediction (NSP). The Masked Language Model task randomly masks some percentage of the input tokens, and then forces the model to predict the masked tokens, similar to a cloze task. The Next Sentence Prediction task asks the model to predict whether one sentence follows the other.

The Next Sentence Prediction task can be used to predict, from the input text (e.g., *"This is the product's first true AI version, and it understands your voice instantly."*), the correct long form (*"Artificial Intelligence"*). Here, the model learns to judge whether the input context that contains the acronym *"AI"* is coherent with the long form *"Artificial Intelligence"*. The Masked Language Model task can memorize the correlation of tokens between the context sequence and long form. Thus, the model learns that the phrase *"Artificial Intelligence"* often co-occurs with *"product"* or *"understand"*.

However, we find that this naive technique does not perform well (see the ablation studies in Table A4). We believe that the reason is that the acronym is usually ambiguous with many candidates (as shown in Table 1), so that the model has difficulties predicting the correct long form by only using the cross-entropy loss of the binary classification. We also observe that the Masked Language Model loss is so small that the model focuses on adapting the Next Sentence Prediction task only.

AcroBERT. To mitigate the weaknesses of the original BERT, we pre-train an adapted BERT, called AcroBERT, by slightly adjusting the Next Sentence Prediction task. The framework is shown in Figure 2. It aims to bring the positive sample pairs closer together, and to push apart the negative sample pairs. We find that already such a simple model can perform very well. For each pair of a candidate long form and a sentence with an acronym, we compose an input for the Next Sentence Prediction task as "[CLS] *long form* [SEP] *sentence* [SEP]". Then, we obtain representations of this sequence by applying BERT to this new input. The final hidden vector $\mathbf{e}^{[CLS]} \in \mathbb{R}^H$ of the first input token ([CLS]) is used as the aggregate representation, where H is the dimension of the hidden vector. Next, the scores for the binary classification are:

$$P(y) = \text{softmax}(\mathbf{e}^{[CLS]}\mathbf{W}), y \in \{0, 1\} \qquad (1)$$

where $\mathbf{W} \in \mathbb{R}^{H \times 2}$ is a trainable matrix initialized with the weights of the original BERT, and the label 0 signifies that this pair of sentences are coherent. We use $d = P(y = 1)$ as the distance between the candidate and the context, and we want the distances of negative pairs to be larger than for positive pairs. For this, we use a triplet loss function that aims to assign higher scores to the correct candidates that match the topic of the input sentence while reducing the scores of irrelevant candidates:

$$\mathcal{L} = \max\left\{0, \lambda - d_{\text{neg}} + d_{\text{pos}}\right\} \qquad (2)$$

where λ is the margin value, and d_{pos} and d_{neg} are the distances for positive and negative pairs, respectively.

The negatives in this triplet framework can be randomly sampled from the dictionary. However, we observe that such random negatives contribute less to the training and result in slower convergence because the initial model can easily distinguish these triplets. Therefore, it is crucial to select harder triplets that are active and beneficial to the training. For this purpose, we introduce a certain number of ambiguous negatives to each mini-batch, e.g., *"Artificial Intelligence"* can be added to the input context as an ambiguous negative sample for the positive pair *"Adequate Intake* [SEP] *In the United States, the AI for potassium for adults is 4.7 grams."* Through the pre-training step, AcroBERT is able to identify the correct long form with the

most consistent theme from numerous candidates based on the input context.

5 Experiments

In this section, we compare AcroBERT empirically to other acronym disambiguation approaches.

5.1 Experimental Settings

Datasets. We use the following datasets for evaluation: Our GLADIS benchmark consists of three subsets covering the General, Scientific, and Biomedical domains. This benchmark is more challenging than prior work due to a large number of ambiguous long forms: each acronym has around 200 candidates on average. We also evaluate AcroBERT on two existing datasets: UAD (Ciosici et al., 2019) and SciAd (Veyseh et al., 2020). They are general and scientific AD datasets, respectively. We reuse the test set of Medmentions but use the UMLS as the target dictionary. We refer to them as **datasets with fewer candidates** because they have fewer candidates per acronym. The average numbers of candidates per acronym are 2.1, 3.1, and 34.2, respectively. See Appendix A.2 for more details on the datasets.

Benchmark Settings. We design two benchmark settings for the unsupervised and supervised scenarios respectively. In the *unsupervised setting*, each model is evaluated on the test sets without access to train and validation sets. In the *fine-tuned setting*, each model is first fine-tuned on train sets and then evaluated on test sets. We focus on the unsupervised setting because it demonstrates that AcroBERT can achieve considerable performances across several domains even without any annotated samples.

Competitors. We compare our approach to the following publicly available competitors: BM25 (Robertson et al., 1995), FastText (Bojanowski et al., 2017), MadDog (Veyseh et al., 2021), BERT (Devlin et al., 2018), BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019). Besides, we introduce a Popularity-Ours baseline that uses the frequency of long forms of our collected pre-training datasets. We do not compare to general entity linking methods, because prior work has already found that general systems like AIDA (Hoffart et al., 2011) tend to lag behind acronym disambiguation models by 10-30 absolute percentage points (Li
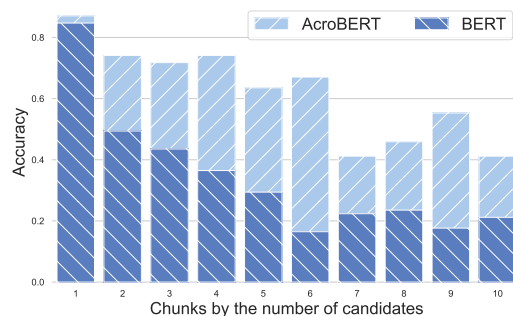

Figure 3: Robustness evaluation of hard samples on the General test set. The samples are divided evenly into ten chunks according to the number of candidates of each sample (results on the other two sets are shown in Figure A1 in the appendix).

et al., 2018). See Appendix A.3 for details on the competitors.

Implementation Details. All approaches are implemented with PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2020) by using one NVIDIA Tesla V100S PCIe 32 GB Specs. For pre-training, we use the parameters in the original BERT to initialize AcroBERT, and then pre-trained on our collected datasets with ~160 million samples for one epoch. The margin of triplet loss is 0.2 and the number of ambiguous negatives is 2 for each mini-batch. See more details in Appendix A.4.

Inference. For the inference stage, every pair of a context sentence and a candidate with the matching short form in the dictionary constitutes an input to the Next Sentence Prediction task. The language model produces a score for each candidate and we select the one with the highest score as the final predicted output.

Metrics. We evaluate the models by precision, recall, and macro F1. These metrics are defined in detail in Section A.5

5.2 Results

5.2.1 Overall Performance

Unsupervised Setting. Table 5 shows the experimental results in the unsupervised setting. We first observe that our AcroBERT significantly outperforms the baselines across the three domains. For example, AcroBERT can improve the original BERT by more than 30 absolute percentage points of F1 on average on our benchmark. Second, the naive popularity method comes second on this

	General				Scientific				Biomedical				Avg	
	Dev		Test		Dev		Test		Dev		Test			
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
BERT (2018)	53.8	70.7	54.9	53.1	13.5	9.9	14.3	10.4	9.8	12.4	9.4	7.5	26.0	27.3
SciBERT (2019)	32.4	38.6	33.6	27.7	22.7	19.4	23.4	17.7	31.2	35.6	31.0	28.3	29.5	27.9
BioBERT (2020)	26.0	23.6	25.7	20.3	11.2	9.7	12.4	9.0	24.0	21.8	20.2	16.8	19.9	16.9
AcroBERT	**72.9**	**76.1**	**71.0**	**76.1**	**28.7**	**34.9**	**29.0**	**27.6**	**62.5**	**62.4**	**60.3**	**69.2**	**54.1**	**57.7**

Table 6: Performances of fine-tuned setting across different models, measured by macro F1 and Accuracy.

benchmark, most likely because it contains a limited number of overshadowed terms. However, it performs badly on the scientific dataset. We assume that this is because this dataset contains 68.7% of overshadowed terms (as shown in Table 3).

Besides, we conduct experiments on existing datasets, namely UAD (Ciosici et al., 2019) and SciAD (Veyseh et al., 2020). Although our method performs consistently well, we relegate this experiment to the appendix B.2 due to the weaknesses of the datasets (small size or data leakage).

Fine-tuned Setting. In this experiment, every pre-trained language model is fine-tuned on the training set by the triplet loss, as introduced in the pre-training step. Negatives are randomly sampled from ambiguous long forms for the correct label, and the results are shown in Table 6. BERT, SciBERT, and BioBERT perform better in their respective fields. However, our AcroBERT achieves the best result across the three fields on average, which demonstrates the effectiveness of the pre-training strategy. One might think that it is unfair that AcroBERT uses the pre-training corpus, while the other models do not. However, there is no other pre-trained model for general disambiguation. Our approach is the first that capitalizes on large-scale corpora and pre-training.

As for the inference speed, AcroBERT has to be run once for every possible long form, which may take some time if there are thousands of long forms, e.g., the acronym AI. However, this runtime can be reduced drastically if one cuts off the less frequent long forms per acronym. Limiting the number of long forms to 23 per acronym, e.g., reduces the worst-case runtime by a factor of 100, while still keeping the recall at 90%.

5.2.2 Robustness Evaluation

Our GLADIS benchmark is more challenging than existing acronym disambiguation datasets due to the much larger acronym dictionary, which means more candidates per acronym. To measure the robustness of acronym disambiguation systems

Model	Popular	Overshadowed	Avg
BERT (2018)	13.3	12.7	13.0
SciBERT (2019)	11.6	8.1	9.9
BioBERT (2020)	2.1	1.0	1.6
AcroBERT	**61.9**	**33.4**	**47.7**

Table 7: Robustness evaluation of overshadowed entities on General test set, measured by Accuracy.

against more candidates, we sort the samples in the dataset in descending order of the number of candidates per acronym, and divide them evenly into 10 chunks. For example, samples in the first chunk have 1.58 candidates on average while that number is 2159 for the last chunk. The experimental results are shown in Figure 3. As expected, the performance of BERT and AcroBERT decreases as the number of candidates increases. The same goes for the other two subsets, as shown in Appendix B.3. However, AcroBERT consistently outperforms BERT on each data chunk, which shows that AcroBERT is able to select the correct long form among the numerous candidates.

Moreover, the challenge with our GLADIS benchmark comes from overshadowed samples, which are harder to disambiguate. To validate the robustness of the models, we divide the General test set into two parts: Popular and Overshadowed, as described in Section 3.2. Next, we compare different language models in the unsupervised setting. As shown in Table 7, our AcroBERT performs best on both the Popular and the Overshadowed subset. We conclude that AcroBERT is more robust against ambiguous and overshadowed samples in acronym disambiguation task.

6 Conclusion

In this paper, we have presented GLADIS, a challenging benchmark for Acronym Disambiguation, which includes a larger dictionary, three datasets from the general, scientific, and biomedical domains, and a large-scale pre-training corpus. We have also proposed AcroBERT, a BERT-based

model that is pre-trained on our collected acronym documents, which can significantly outperform other baselines across multiple domains, and which is more robust in the presence of very ambiguous acronyms and overshadowed samples. For future work, we aim to enhance the performance of AcroBERT on the overshadowed cases, which is crucial for the acronym disambiguation task.

7 Limitations

We see two main limitations of our work. First, although the current acronym dictionary is of relatively high quality, it still contains a small fraction of duplicate long forms due to typos (as in *"Convlutional Neural Network"*), morphological changes (as in *"Convolutional Neuronal Network"*) and nested acronyms (as in *"convolutional NN"*). A manual evaluation of 100 randomly chosen long forms from the three datasets in GLADIS shows that 6% of them are noisy. At the same time, the frequency of these noisy forms is much lower than that of the standard long forms: all noisy forms in the sample taken together appear 100 times in the corpus – compared to 31k times for the clean forms. Thus, the percentage of clean forms, weighted by their frequency, is 97%. A good AD system should select the most frequent one among noisy forms for an acronym, and in our sample none of the most frequent forms was noisy.

A second limitation of our approach is that the performance of the current AcroBERT system on the Scientific dataset still needs improvement. We are considering to introduce more pre-training data from this domain to address this issue.

Ethics Statement

This work presents GLADIS, a free and open benchmark for the research community to study Acronym Disambiguation, which consists of three components: a dictionary, a pre-training corpus, and three domain-specific datasets. The dictionary and pre-training corpus are collected from the Pile dataset (Gao et al., 2020), which is a public dataset under the MIT license. The three domain-specific datasets are adapted from SciAD (Veysseh et al., 2020), WikilinksNED Unseen Mentions (Onoe and Durrett, 2020) and Medmentions (Mohan and Li, 2018), respectively. They all allow sharing and redistribution. The source datasets and their publications will be credited on our GitHub page, and their licenses will be mentioned both on the Web page and in the downloads of GLADIS.

References

Eytan Adar. 2004. Sarad: A simple and robust abbreviation dictionary. *Bioinformatics*, 20(4):527–533.

Hiroko Ao and Toshihisa Takagi. 2005. Alice: an algorithm to extract abbreviations from medline. *Journal of the American Medical Informatics Association*, 12(5):576–586.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Jeffrey T Chang, Hinrich Schütze, and Russ B Altman. 2002. Creating an online dictionary of abbreviations from medline. *Journal of the American Medical Informatics Association*, 9(6):612–620.

Jean Charbonnier and Christian Wartena. 2018. Using word embeddings for unsupervised acronym disambiguation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2610–2619.

Manuel Ciosici, Tobias Sommer, and Ira Assent. 2019. Unsupervised abbreviation disambiguation contextual disambiguation using word embeddings. *arXiv preprint arXiv:1904.00929*.

Manuel R Ciosici and Ira Assent. 2018. Abbreviation expander-a web-based system for easy reading of technical documents. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 1–4.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

István Endrédy and Attila Novák. 2013. More effective boilerplate removal-the goldminer algorithm. *Polibits*, (48):79–83.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named entity disambiguation for noisy text. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 58–68.

Gregory P Finley, Serguei VS Pakhomov, Reed McEwan, and Genevieve B Melton. 2016. Towards comprehensive clinical abbreviation disambiguation using machine-labeled training data. In *AMIA Annual Symposium Proceedings*, volume 2016, page 560. American Medical Informatics Association.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Mauricio Girardi-Schappo, Ludmila Brochini, Ariadne A Costa, Tawan TA Carvalho, and Osame Kinouchi. 2019. Self-organized critical balanced networks: a unified framework. *arXiv preprint arXiv:1906.05624*.

Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravičius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of biomedical semantics*, 5(1):1–25.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 782–792.

Rezarta Islamaj Dogan, G Craig Murray, Aurélie Névéol, and Zhiyong Lu. 2009. Understanding pubmed® user search behavior through log analysis. *Database*, 2009.

Alpa Jain, Silviu Cucerzan, and Saliha Azzam. 2007. Acronym-expansion recognition and ranking on the web. In *2007 IEEE International Conference on Information Reuse and Integration*, pages 209–214. IEEE.

Qiao Jin, Jinling Liu, and Xinghua Lu. 2019. Deep contextualized biomedical abbreviation expansion. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 88–96.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86.

Cheng-Ju Kuo, Maurice HT Ling, Kuan-Ting Lin, and Chun-Nan Hsu. 2009. Bioadi: a machine learning approach to identifying abbreviations and definitions in biological literature. In *BMC bioinformatics*, volume 10, pages 1–10. BioMed Central.

Leah S Larkey, Paul Ogilvie, M Andrew Price, and Brenden Tamilio. 2000. Acrophile: an automated acronym extractor and server. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 205–214.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Bin Li, Fei Xia, Yixuan Weng, Xiusheng Huang, Bin Sun, and Shutao Li. 2021. Simclad: A simple framework for contrastive learning of acronym disambiguation. *arXiv preprint arXiv:2111.14306*.

Chao Li, Lei Ji, and Jun Yan. 2015. Acronym disambiguation using word embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Yang Li, Bo Zhao, Ariel Fuxman, and Fangbo Tao. 2018. Guess me if you can: Acronym disambiguation for enterprises. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1308–1317.

Jie Liu, Caihua Liu, and Yalou Huang. 2017. Multigranularity sequence labeling model for acronym expansion identification. *Information Sciences*, 378:462–474.

Sunil Mohan and Donghui Li. 2018. Medmentions: A large biomedical corpus annotated with umls concepts. In *Automated Knowledge Base Construction (AKBC)*.

Dana Movshovitz-Attias and William Cohen. 2012. Alignment-hmm-based extraction of abbreviations from biomedical text. In *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 47–55.

David Nadeau and Peter D Turney. 2005. A supervised learning approach to acronym identification. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 319–329. Springer.

Naoaki Okazaki and Sophia Ananiadou. 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(24):3089–3095.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8576–8583.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*.

Serguei Pakhomov, Ted Pedersen, and Christopher G Chute. 2005. Abbreviation and acronym disambiguation in clinical discourse. In *AMIA Annual Symposium Proceedings*, volume 2005, page 589. American Medical Informatics Association.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. Bert-based acronym disambiguation with multiple training strategies. *arXiv preprint arXiv:2103.00488*.

Youngja Park and Roy J Byrd. 2001. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 conference on empirical methods in natural language processing*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *European Semantic Web Conference*, pages 583–596. Springer.

Vera Provatorova, Samarth Bhargav, Svitlana Vakulenko, and Evangelos Kanoulas. 2021. Robustness evaluation of entity disambiguation using prior probes: the case of entity overshadowing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10501–10510.

James Pustejovsky, José Castano, Brent Cochran, Maciej Kotecki, and Michael Morrell. 2001. Automatic extraction of acronym-meaning pairs from medline databases. In *MEDINFO 2001*, pages 371–375. IOS Press.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Stephen E Robertson, Steve Walker, Susan Jones, et al. 1995. Okapi at trec-3.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2018. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Ariel S Schwartz and Marti A Hearst. 2002. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, pages 451–462. World Scientific.

Sunghwan Sohn, Donald C Comeau, Won Kim, and W John Wilbur. 2008. Abbreviation definition identification based on automatic precision estimates. *BMC bioinformatics*, 9(1):1–10.

Mark Stevenson, Yikun Guo, Abdulaziz Alamri, and Robert Gaizauskas. 2009. Disambiguation of biomedical abbreviations. In *Proceedings of the BioNLP 2009 Workshop*, pages 71–79.

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R South, Danielle L Mowery, Gareth JF Jones, et al. 2013. Overview of the share/clef ehealth evaluation lab 2013. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 212–231. Springer.

Jörg Tiedemann. 2016. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3518–3522.

Karin Vadovičová. 2014. Affective and cognitive prefrontal cortex projections to the lateral habenula in humans. *Frontiers in human neuroscience*, 8:819.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Walter Chang, and Thien Huu Nguyen. 2021. Maddog: A web-based system for acronym identification and disambiguation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 160–167.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, and Thien Huu Nguyen. 2020. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3285–3301.

Amir Pouran Ben Veyseh, Nicole Meister, Seunghyun Yoon, Rajiv Jain, Franck Dernoncourt, and Thien Huu Nguyen. 2022. Macronym: A large-scale dataset for multilingual and multi-domain acronym extraction. *arXiv preprint arXiv:2202.09694*.

Zhi Wen, Xing Han Lu, and Siva Reddy. 2020. Medal: Medical abbreviation disambiguation dataset for natural language understanding pretraining. *arXiv preprint arXiv:2012.13978*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yonghui Wu, Joshua C Denny, S Trent Rosenbloom, Randolph A Miller, Dario A Giuse, Lulu Wang, Carmelo Blanquicett, Ergin Soysal, Jun Xu, and Hua Xu. 2017. A long journey to short abbreviations: developing an open-source framework for clinical abbreviation recognition and disambiguation (card). *Journal of the American Medical Informatics Association*, 24(e1):e79–e86.

Yonghui Wu, Jun Xu, Yaoyun Zhang, and Hua Xu. 2015. Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of BioNLP 15*, pages 171–176.

Stuart Yeates, David Bainbridge, and Ian H Witten. 2000. Using compression to identify acronyms in text. In *Proceedings of the Conference on Data Compression*, page 582.

Hong Yu, George Hripcsak, and Carol Friedman. 2002. Mapping abbreviations to full forms in biomedical articles. *Journal of the American Medical Informatics Association*, 9(3):262–272.

Hong Yu, Won Kim, Vasileios Hatzivassiloglou, and W John Wilbur. 2007. Using medline as a knowledge source for disambiguating abbreviations and acronyms in full-text biomedical journal articles. *Journal of biomedical informatics*, 40(2):150–159.

Qiwei Zhong, Guanxiong Zeng, Danqing Zhu, Yang Zhang, Wangli Lin, Ben Chen, and Jiayu Tang. 2021. Leveraging domain agnostic and specific knowledge for acronym disambiguation. *arXiv preprint arXiv:2107.00316*.

Danqing Zhu, Wangli Lin, Yang Zhang, Qiwei Zhong, Guanxiong Zeng, Weilin Wu, and Jiayu Tang. 2021. At-bert: Adversarial training bert for acronym identification winning solution for sdu@ aaai-21. *arXiv preprint arXiv:2101.03700*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Details of the Experimental Settings

A.1 Details of the Pile Dataset

Pile (Gao et al., 2020) is an 825 GiB English text corpus designed to train large-scale language models, which is constructed from 22 diverse high-quality academic or professional sources. Pile is constructed from existing or newly introduced datasets, and we present these sources here. **Pile-CC** is a collection of web pages, metadata and texts, which is extracted from jusText (Endrédy and Novák, 2013). **Books3** is a book dataset of fiction and nonfiction books derived from Bibliotik [1]. **Project Gutenberg** has classic Western literature derived from PG-19 (Rae et al., 2019). **OpenSubtitles** provides a large corpus of English subtitles from movies and television shows collected by Tiedemann (2016). **DeepMind Mathematics** is a collection of many different types of mathematics questions (Saxton et al., 2018). **BookCorpus2** is an expanded version of BookCorpus (Zhu et al., 2015), a corpus of books from the web. **EuroParl** is a corpus of parallel text in 11 languages from the proceedings of the European Parliament(Koehn, 2005). **Enron Emails** is a large set of email messages, which contains 619,446 messages belonging to 158 users (Klimt and Yang, 2004).

A.2 Details of the Experimental Datasets

We use the following benchmarks for Acronym Disambiguation:

Our GLADIS benchmark consists of three subsets covering the General, Scientific, and Biomedical domains. It is a very challenging benchmark, due to a large number of ambiguous long forms, as described in Section 3.2.

- **General** has 45K samples gathered from the WikilinksNED Unseen Mentions (Onoe and Durrett, 2020).

- **Scientific** is adapted from SciAD (Veyseh et al., 2020) with 56K samples, and the long forms in the original dataset are mapped to the new acronym dictionary. We re-split the training, validation and test sets to assure there are no overlaps.

- **Biomedical** includes 12K samples obtained from Medmentions (Mohan and Li, 2018).

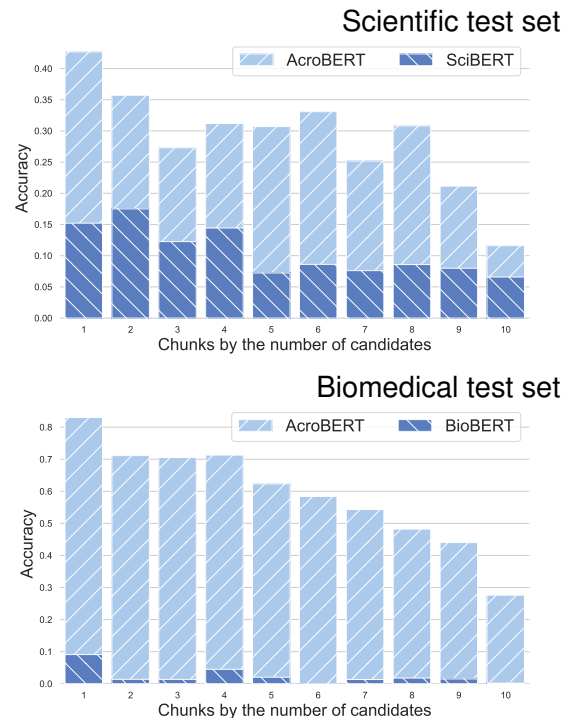[1] https://twitter.com/theshawwn/status/1320282149329784833

Figure A1: Robustness evaluation of hard samples on Scientific and Biomedical test set. The samples are divided evenly into ten chunks according to the number of candidates of each sample.

Datasets with fewer candidates per acronym.

- **UAD** (Ciosici et al., 2019) is gathered from the English Wikipedia and we use the manually labeled 7K samples for evaluation.

- **SciAD** (Veyseh et al., 2020) is a human-annotated dataset for the scientific domain with 62K samples gathered from the ArXiv preprint papers, and the validation set with 6K samples is used for experiments.

- **Biomedical-UMLS** is a dataset with 3K samples obtained from the test set in our benchmark by using the UMLS concepts as the acronym dictionary

The average candidates per acronym for the three datasets are 2.1, 3.1, and 34.2, respectively.

A.3 Competitors

We compare our approach to the following publicly available competitors:

- BM25 (Robertson et al., 1995) is a classical ranking function in information retrieval.

Model	UAD		SciAD		Biomedical-UMLS		Avg	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
BERT (2018)	**89.3**	91.1	54.1	57.2	38.0	32.7	60.5	60.3
SciBERT (2019)	74.8	78.4	**65.6**	71.7	54.9	50.3	65.1	66.8
BioBERT (2020)	66.2	68.2	19.7	22.4	37.4	31.4	41.1	40.7
AcroBERT	88.8	**93.7**	58.0	**72.0**	**67.5**	**65.3**	**71.4**	**77.0**

Table A1: Performances on benchmarks with fewer candidates, measured by macro F1 and Accuracy.

- Popularity-Ours is a baseline that uses the frequency of long forms of our collected pre-training datasets.

- BERT (Devlin et al., 2018) is a strong baseline, which pre-trains contextual language models on large corpora. The scores for the NSP task can be used for the acronym disambiguation.

- FastText (Bojanowski et al., 2017) is a character-level embeddings and can produce representations for arbitrary words. In this experiment, we first represent the input sentence and candidates by the sum of word embeddings from FastText. Then, all candidates are ranked by their cosine similarity score.

- MadDog (Veyseh et al., 2021) is a web-based acronym disambiguation system for multiple domains. It first creates chunks in which all samples with the same acronyms are assigned to the same chunks. After, a separate Bi-LSTM model is trained for each chunk. To deploy the MadDog server, it needs at least 125 GB of disk space and 70 GB of RAM memory [2].

- BioBERT (Lee et al., 2020) is a biomedical language representation model mainly pre-trained on PubMed Abstracts and PMC Full-text articles, which is a strong baseline in the biomedical domain.

- SciBERT (Beltagy et al., 2019) is a scientific language model based on BERT pre-trained on a large multi-domain corpus of scientific publications, which can improve performance on downstream scientific NLP tasks.

A.4 Implementation Details

All approaches are implemented with PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al.,

2020). When pre-training AcroBERT, the model is initialized by the parameters in the original BERT, and then pre-trained on our collected datasets for one epoch. In total, there are ~160 million samples in this corpus, covering various domains. The batch size is 32, and we use Adam (Kingma and Ba, 2014) with a learning rate $2e-5$ for optimization. The learning rate is exponentially decayed for every 10,000 steps with a rate of 0.95. The margin of triplet loss is 0.2 and the number of ambiguous negatives is 2 for each mini-batch.

For the fine-tuning stage, each competitor model is initialized with the pre-trained parameters from HuggingFace, and we use AcroBERT after pre-training for comparison. All models are fine-tuned by using the Triplet loss. All parameters of each model are fine-tuned in this experiment, across all domains by using the same hyper-parameters. The batch size is 8 and the learning rate is among $[1e-5, 8e-6, 6e-6, 4e-6, 2e-6]$ for the Adam optimizer. The model that has the best performance on the validation set among the 5 learning rates is evaluated on the test set. We use one NVIDIA Tesla V100S PCIe 32 GB Specs.

A.5 Metrics

Acronym disambiguation can be seen as a classification problem, where the input is (1) a dictionary of acronyms and (2) a sentence with an acronym. Each long form for that acronym from the dictionary is considered a class, and the acronym disambiguation has to choose the correct class. We evaluate the models by precision, recall, and macro F1. There are two ways to calculate the macro F1: "F1 of Averages" and "Averaged F1". The first computes the F1 value over the arithmetic means of precision and recall, while the second computes the F1 value for each class, and then averages them. Some prior works adopt the first method. However, this method gives a higher weight to popular classes, and it may thus unfairly yield a high score if the model works well on these popular classes

---

[2]https://github.com/amirveyseh/MadDog

2086

Acronym	Long Form	Provenance
ELEC	*Election Law Enforcement Commission*	*Christie, some legislators and the state Election Law Enforcement Commission (**ELEC**), have joined the comptroller in voicing support for the elimination of the loophole.*
ISR	*in-stent restenosis*	*Although conventional stents are routinely used in clinical procedures, clinical data shows that these stents are not capable of completely preventing in-stent restenosis (**ISR**) or restenosis caused by intimal hyperplasia.*
IL-6	*interleukin-6*	*Consistent blood markers in afflicted patients are normal to low white cell counts and elevated interleukin-6 (**IL-6**) levels which, among its many activities, signal the liver to increase synthesis and secretion of CRP.*
PCP	*Planar cell polarity*	*Establishment of photoreceptor cell polarity and ciliogenesis Planar cell polarity (**PCP**)-associated Prickle genes (Pk1 and Pk2) are tissue polarity genes necessary for the establishment of PCP in Drosophila.*
DEP	*dielectrophoretic*	*They included: a particle counter, trypan blue exclusion (Cedex), an in situ bulk capacitance probe, an off-line fluorescent flow cytometer, and a prototype dielectrophoretic (**DEP**) cytometer.*
AQP3	*aquaporin3*	*The laxative effect of bisacodyl is attributable to decreased aquaporin-3 expression in the colon induced by increased PGE2 secretion from macrophages. The purpose of this study was to investigate the role of aquaporin3 (**AQP3**) in the colon in the laxative effect of bisacodyl.*

Table A2: Samples of extracted acronyms, long forms and provenances by using the rule-based algorithm from Schwartz and Hearst (2002).

Acronym	Long Form	Context	BERT	AcroBERT
ECB	*European Central Bank*	*Being made to bring the main road network in Romania in the European corridors. There have been initiated several projects to modernize the network of **ECB** corridors, financed from ispa funds and state-guaranteed loans from international financial institutions. Government seeks external financing or public-private partnerships for other road network upgrades , especially*	*External Commercial Borrowing*	*European Central Bank*
PR	*Public Relations*	*A whistleblower like monologist Mike Daisey gets targeted as a scapegoat who must be discredited and diminished in the public 's eye. More often than not, **PR** is a preemptive process. Celebrity publicists are paid lots of money to keep certain stories out of the news.*	*Preemptive-Resume*	*Public Relations*
PUD	*Peptic Ulcer Disease*	*Tumors cause an overactivation of these hormone-producing glands, leading to serious health problems such as severe **PUD** (due to gastrin hypersecretion, which stimulates secretion of hydrochloric acd).*	*Psychogenic Urinary Dysfunction*	*Peptic Ulcer Disease*
WFC	*Walsall F.C.*	*Injury during a game against Norwich city on the 13 march 2010, forcing him to miss Huddersfields next five games. He made his return against **WFC** on the 13 April 2010 , coming on as a 75th minute substitute and scoring a stoppage time winner to make the score 4a3 to town.*	*Wide Free Choice*	*World Fighting Championships*

Table A3: Case study of predicted results by BERT and AcroBERT.

only (Opitz and Burst, 2019). Therefore, we use the Averaged F1 across classes as our metric, which is more robust towards the error type distribution. That is:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, i \in \{1, 2, ..., n\} \quad \text{(A.1)}$$

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, i \in \{1, 2, ..., n\} \quad \text{(A.2)}$$

$$\text{F1} = \frac{1}{n} \sum_{i=1}^{n} \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad \text{(A.3)}$$

B Additional Experiments

B.1 Ablation Study

In this experiment, we validate the effectiveness of the pre-training strategy in AcroBERT, which adopts a triplet loss with negative samples from ambiguous candidates. Every model is initialized with the parameters of the original BERT, and we use various strategies for pre-training: only Masked Language Model, only Next Sentence Prediction, and the combination of the two and the triplet framework in AcroBERT. Each strategy is pre-trained on our collected corpus for 300K steps and then the corresponding model is evaluated on the three validation sets. The results (in Table A4) show that the strategy of AcroBERT is most beneficial for the acronym disambiguation, as it performs the best on average. The Next Sentence Prediction task is more important than the Masked Language Model task. Even if MLM is not used, the impact on the model is not significant, which means the

Strategy	General		Scientific		Biomedical		Avg	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Triplet framework in AcroBERT	**61.2**	**64.2**	**18.8**	**18.4**	9.1	10.1	**29.7**	**30.9**
MLM + NSP of the original BERT	45.7	55.7	15.8	14.9	**10.2**	**11.4**	23.9	27.3
only NSP	52.4	58.7	15.1	12.4	6.1	6.5	24.5	25.9
only MLM	11.2	14.4	2.7	3.5	1.4	1.5	5.1	6.5

Table A4: Ablation studies for different pre-training strategies after 300K steps across three validation sets, measured by macro F1 and Accuracy.

original BERT has already learned it well.

B.2 Experiments on Benchmarks with Fewer Candidates

As mentioned before, one drawback of the prior AD benchmark is that the magnitude of the acronym dictionary is small, which is not consistent with practical applications. In this experiment, we therefore valid the performance of AcroBERT on datasets with fewer candidates. The results are shown in Table A1, and we observe that AcroBERT can achieve the best average scores across three datasets again, which demonstrates the generalization capability of our AcroBERT. On the other hand, the lead of our model is not as substantial as before. This is because there are fewer candidates per acronym, and AcroBERT is particularly well-suited for identifying the correct one among a large number of candidates.

B.3 Robustness Evaluation for Many Candidates

Similar to Section 5.2.2, we analyse the robustness of AcroBERT on the other two domains. Each test set is divided evenly into 10 chunks by the number of candidates. The first chunk has the least number of candidates while the last chunk has the most, up to more than 2K. Figure A1 shows the experimental scores on the Scientific and Biomedical test set. We observe that for the first chunk, SciBERT and BioBERT are on par with our AcroBERT. However, AcroBERT outperforms the two significantly when the number of candidates get larger.

B.4 Case Study

Table A3 shows case studies of the outputs by BERT and AcroBERT. BERT often uses the memorized correlations of tokens for reasoning and this can cause errors. For example, *External Commercial Borrowings* are loans in India made by non-resident lenders in foreign currency to Indian bor-

rowers [3]. BERT can determine this correct long form probably with help of the key phrase *"external financing"*. For the third case, *Peptic Ulcer Disease* is more consistent with the input context. However, BERT fails on it while AcroBERT benefits from the pre-training strategy and is able to distinguish different candidates based on contexts. For the fourth sample, both methods fail, most likely because of the low frequency of the long forms and the uninformative contexts.

---

[3]https://en.wikipedia.org/wiki/External_commercial_borrowing