Weakly Supervised Monotonic Character Alignment for Acronym-Long-Form Mapping

Anonymous ACL submission

Abstract

Aligning acronyms to their long forms is a critical but underexplored problem in entity resolution and text retrieval. There are many ways a long form can be compressed into its acronym (e.g. initials, syllabus, partial words) and character-level annotations are rarely available in real-world data. To address these challenges, we present a weakly supervised approach that formulates acronym-long-form mapping as a monotonic character subsequence alignment task. First, we generate weak alignment labels by combining positional weights with a beam-search decoder. Next, the weak labels are used to train a character-level sequence labeller that predicts, for each long-form character, the likelihood that it is part of the acronym. During inference, we perform a secondary beam search over the character-level scores to recover the most probable acronym-long-form mapping. Based on our experiments on three datasets curated from publicly available sources, our approach outperforms heuristic baselines. Additionally, it achieves comparable performance to variants trained on weak labels generated by large language models (LLMs), while requiring substantially less compute. This underscores its efficacy for low-resource, real-world environments.

1 Introduction

In the context of entity resolution and text retrieval, aligning long-form text to their shorthand notation is critical for identifying, linking and deduplicating semantically similar entities. Humans frequently compress long text into short forms by retaining key characters from the original text. For instance, the short form *IBM* (*International Business Machines*) is constructed by taking the first letter of each word, whereas *FedEx* (*Federal Express*) combines the first three characters of the first word with the first two characters of the second. In this work, we use the term acronym to refer to short forms such as abbreviations and initialisms.

A fundamental challenge arises when there are multiple plausible ways an acronym can be constructed from its long form. For example, for the organization *Midcontinent Independent System Operator* (*MISO*), there are at least two ways to derive *MISO*: (i) taking the first two letters of the first word **Mi**dcontinent Independent **S**ystem **O**perator, or (ii) taking the first letter of each word **M**idcontinent **I**ndependent **S**ystem **O**perator. This variability introduces ambiguity, making acronym-long-form mapping non-trivial.

In addition, obtaining acronym-long-form mappings presents additional challenges. Manual character-level alignment is a time-consuming, labour intensive process, and alignment signals in free-text documents are sparse and noisy. Acronyms may appear in parentheses after their long form (e.g. Hewlett-Packard (HP)) or their corresponding character may be typographically emphasized (e.g. **N**ational **B**roadcasting **C**ompany). However, in practice, such indicators may be lost during case folding, a common step in text preprocessing and deduplication pipelines (Gyawali et al., 2020).

These challenges limit the effectiveness of entity resolution and text retrieval systems in high-stakes compliance and finance domains, such as Anti Money Laundering (AML) screening, where matching individuals and corporate entities against sanctions and watchlists is critical. Missed matches (false negatives) exposes institutions to regulatory risk, while incorrect matches (false positives) create unnecessary alerts and operational overhead (Oliveira and Leal, 2025; Oztas et al., 2024). Therefore, such systems are expected to reliably match acronyms and long forms of corporate entities in customer records against sanctions or watchlists.

While large language models (LLMs) proved effective for acronym-related tasks such as extracting acronyms and identifying short-long-form expansions (Ali et al., 2024; Kugic et al., 2025), relying

on them in high-stakes compliance and finance settings raises concerns around latency, privacy and computational cost (Patel et al., 2025; Yang et al., 2025). In this work, we focus on a lightweight approach designed for resource constrained or air-gapped compliance environments where hosting or querying third-party LLMs is not feasible.

To achieve this, we model the task of mapping an acronym to its corresponding long form as a monotonic character alignment problem. An acronym is viewed as a compressed subsequence of its long form that retains a subset of informative characters. Motivated by the observation that acronym characters often align to early positions within words in the long form, we propose a lightweight weak-label generation approach that combines the position of characters in each word with a monotonic beam search decoder to generate weak alignment labels. The weak labels are used to train a character-level sequence labelling (CLSL) model to predict, for each character in the long form, the probability that it is part of the acronym. During inference, the scores generated by the CLSL model are combined with a secondary monotonic beam search decoder to obtain the highest scoring acronym-long-form mapping.

In summary, the key contributions of this work are as follows:

- **Positional Decay Alignment (PDA) for weak label generation**: We introduce PDA, a lightweight heuristic aligner that assigns higher weights to earlier characters in each word within the long form, combined with a monotonic beam-search decoder to select the highest-scoring acronym-long-form subsequence. PDA produces character-level alignments without manual annotation or external language models.

- **A weakly supervised framework for monotonic character alignment**: We propose a two-stage framework that: (i) trains a character-level sequence labelling model on weak labels generated by PDA and (ii) during inference, combines the model scores with a secondary monotonic beam-search decoder to obtain acronym-long-form alignments. Our weakly supervised approach achieves exact-match accuracy comparable to variants trained on LLM-generated labels, while remaining lightweight and self-contained.

- **Acronym Matching (AM) benchmarks and baselines**: We adapt three publicly available datasets (corporate names, arXiv titles and scientific terminology) into AM benchmarks and provide validated character-level alignments for each test set. We release these datasets together with the baseline implementations to facilitate future work.

The remainder of this paper is structured into several key sections: Section 2 reviews existing work that establishes the relationship between acronym and long form text. Section 3 presents our proposed methodology. Section 4 details the results of our experiments, Section 5 concludes with a summary of our findings and Section 6 discuses the limitations and future directions.

2 Related Work

This section provides an overview of previous research on processing acronym and long form text. Prior work primarily focused on two tasks: Acronym Identification (AI) and Acronym Disambiguation (AD). AI aims to extract acronyms and their corresponding long forms from free-text, whereas AD selects the correct long form for an acronym based on context. In this work we refer to the task of aligning an acronym to its long form as Acronym Matching (AM). Table 2 summarizes the three acronym processing tasks.

For AI, Schwartz and Hearst (2003) introduce a greedy heuristic approach to extract acronyms and their corresponding long forms. This approach assumes that acronym appear in parenthesis after their long form: `<long form> (<acronym>)` and maps acronym characters to the long form in a greedy manner. Veyseh et al. (2021) builds on this approach and propose an extended set of rules that handle cases where the capitalised initials of the long form match the acronym. Egan and Bohannon (2020) formulates AI as a sequence labelling task where a transformer-based encoder, XLNet (Yang et al., 2019), followed by a linear classification layer is used to predict the token-level boundaries of acronyms and long forms.

Kugic et al. (2025); Egan and Bohannon (2020) model AD as an information retrieval problem: given a sentence containing an acronym, the goal is to retrieve the most similar sentence in the reference lookup and use its tagged label for prediction. Kugic et al. (2025) leverage the GPT-4o LLM (Achiam et al., 2023) to generate synthetic clini-

Task	Description	Example
Acronym Identification (AI)	Extract acronyms and their long forms from text.	"We use the Gramian Angular Field (GAF) to ..." → (GAF, Gramian Angular Field) (Veyseh et al., 2020)
Acronym Disambiguation (AD)	Given a sentence containing an acronym, identify the correct expansion.	"... bovine liver **DHF** reductase ..." → **dihydrofolate** (Wen et al., 2020)
Acronym Matching (AM)	Determine the character-level mapping between an acronym and its long form.	(IBM, International Business Machines) → **I**nternational **B**usiness **M**achines

Table 1: Summary of acronym-related tasks for Acronym Identification (AI), Acronym Disambiguation (AD) and Acronym Matching (AM). Acronym, long form pairs are denoted as (`<acronym>`, `<long form>`), input and output as `<input>` → `<output>`, and for AM, bold characters indicate the letters selected to form the acronym.

cal narratives based on acronym-long-form pairs. The generated text are encoded with a BERT-based model, MedBERT.de (Bressem et al., 2023), and indexed with a vector database for nearest-neighbour search based on cosine similarity. Likewise, Egan and Bohannon (2020) leverage transformer-based encoders to select the label with the highest cosine similarity score between a test example its reference lookup.

However, prior work in AI and AD does not effectively model the character-level alignments between acronyms and their long form. Transformer-based AI and AD approaches typically rely on subword tokenization, overlooking the fine-grained acronym-long-form mappings required for AM. In addition, approaches such as Schwartz and Hearst (2003); Veyseh et al. (2021) rely on greedy heuristics that may fail when multiple plausible subsequences exist in the long form. This highlights the need for a robust weakly supervised character-level alignment framework for AM.

3 Method

In this section, we present our weakly supervised monotonic alignment framework to obtain character-level mappings between acronyms and their long forms. As illustrated in Figure 1, our approach comprises three key modules:

1. **Positional Decay Alignment (PDA)**: A heuristic aligner that (i) assigns higher weights to earlier characters in each word of the long form and (ii) uses a monotonic beam-search decoder to select the highest scoring subsequence. The resulting character-level alignments are used as weak labels.

2. **Character-level Sequence Labelling (CLSL) Model**: A character-level sequence labelling model trained on weak labels generated by PDA. The model predicts, for each character, the likelihood that it is an acronym character.

3. **Beam-search Aligner**: During inference, the scores generated by the CLSL model are combined with a secondary monotonic beam-search decoder, which searches over valid subsequences and returns the highest-scoring acronym-long-form alignment.

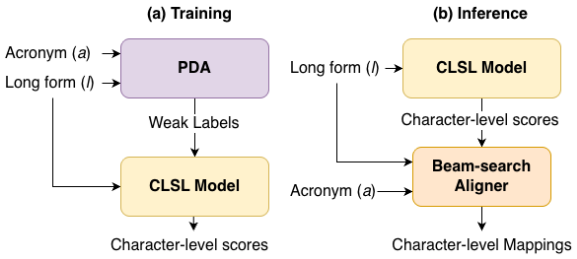

Figure 1: Overall architecture of our proposed weakly supervised monotonic alignment framework for Acronym Matching (AM).

3.1 Positional Decay Alignment (PDA)

We model acronyms as compressed subsequences of their long forms that retain a subset of informative characters. We observe that acronym characters frequently occur near the beginning of words in their long form. Therefore, to capture this pattern, we introduce our Positional Decay Alignment (PDA) approach, which assigns decaying positional weights to characters within each word and a monotonic beam search decoder that selects the subsequence with the highest cumulative score.

Given a long form $l = (l_1, \ldots, l_m)$, we assign a positional weight w_j to each character l_j using the exponentially decaying function detailed in Equation 1 where p_j refers to the position of l_j within each word and $\lambda > 0$ controls the rate of decay. For example, when $\lambda = 0.01$, the positional weights for the long form *Korea Telecom* (separated by the

Task	Description	Example
Acronym Identification (AI)	Extract acronyms and their long forms from text.	"We use the Gramian Angular Field (GAF) to ..." → (GAF, Gramian Angular Field) (Veyseh et al., 2020)
Acronym Disambiguation (AD)	Given a sentence containing an acronym, identify the correct expansion.	"... bovine liver **DHF** reductase ..." → **dihydrofolate** (Wen et al., 2020)
Acronym Matching (AM)	Determine the character-level mapping between an acronym and its long form.	(IBM, International Business Machines) → **I**nternational **B**usiness **M**achines

Table 2: Summary of acronym-related tasks. Examples are organized as *input* → *output* and short-long-form pairs are denoted as (short-form, long-form). For AM, bold characters indicate the letters selected to form the acronym.

word delimiter ' ') are: ('K', 1.00), ('o', 0.99), ('r', 0.98), ..., ('T', 1.00), ('e',0.99), ('l',0.98), ...

$$w_j = e^{-\lambda(p_j - 1)}, \tag{1}$$

To obtain character-level acronym-long-form alignments, we combine the positional weights with a monotonic decoder. Given an acronym $a = (a_1, \ldots, a_n)$, its long form $l = (l_1, \ldots, l_m)$ and the positional weights $w = (w_1, \ldots, w_m)$, we search across monotonic subsequences of l where characters $l_{i_k} = a_k$ for $k = 1, \ldots, n$. The optimal alignment is the subsequence that maximises the cumulative positional weights defined in Equation 2.

$$\begin{aligned} p^* &= \arg\max_{p \in \mathcal{P}(a,l)} \sum_{k=1}^{n} w_{i_k}, \\ \mathcal{P}(a,l) &= \{(i_1, \ldots, i_n) : 1 \le i_1 < \cdots < i_n \le m, \\ &\qquad l_{i_k} = a_k \; \forall k = 1, \ldots, n\}. \end{aligned} \tag{2}$$

In practice, we approximate this objective with a monotonic beam-search decoder that returns the indices $p^* = (i_1^*, \ldots, i_n^*)$. Next, we encode p^* as binary labels $y = (y_1, \ldots, y_m)$ where $y_j = 1$ if $j \in p$ and $y_j = 0$ otherwise. This approach approximates global alignment and overcomes the limitations of greedy heuristics that only consider the first matching subsequence. Further details of our beam-search formulation are provided in the *Beam-search Aligner* subsection.

3.2 Character-level Sequence Labelling Model

Given the weak labels generated by PDA, we train a character-level model that learns which characters in the long form are likely to be used in an acronym. We formulate this as a sequence labelling task, defined in Equation 3, where the model takes the long-form character sequence $l = (l_1, \ldots, l_m)$ as input and for each position j, predicts the probability that character l_j is part of the acronym.

$$P_\theta(y \mid l) = \prod_{j=1}^{m} P_\theta(y_j \mid l), \tag{3}$$

This allows us to utilise character or byte-level encoders such as CANINE (Clark et al., 2022) and ByT5 (Xue et al., 2022) to obtain contextual representations for each character, followed by a linear classification layer that predicts a binary label for each position. We train the model by minimizing the binary cross-entropy loss between the predictions and weak PDA labels. Although PDA relies primarily on positional information and can assign incorrect labels in ambiguous cases, in our experiments the learned model demonstrates contextual capabilities, assigning higher scores to characters in more informative words than those in stopwords.

3.3 Beam-search Aligner

To align an acronym to its long form when multiple valid subsequences exist, a direct approach is to iterate through all subsequences and select the one that maximizes the alignment score in Equation 2. However, this exhaustive search evaluates low-scoring partial alignments that could be discarded early without affecting the final result. For example, given a = *HUD* and l = *Housing and Urban Development*, for the partial acronym ***HU***, a partial alignment such as ***Hou****sing and Urban Development* can be safely discarded once a higher scoring alternative, like ***H****ousing and* ***U****rban Development*, is observed. Therefore, we approximate the global alignment with beam search, which only retains the top-K partial alignments at each step as opposed to an exhaustive search.

We represent a partial alignment as a state (k, j), where the first k characters of the acronym $a = (a_1, \ldots, a_n)$ have been aligned to the first j characters in the long form $l = (l_1, \ldots, l_m)$ and the last matched character $a_k = l_j$. Each state tracks the cumulative alignment score of the partial path. From the state (k, j) where $k < n$, we extend the

partial alignment by matching the next acronym character a_{k+1}. To achieve this, beam search considers all positions $j' > j$ in the long form such that $l_{j'} = a_{k+1}$, and creates a new state $(k+1, j')$. At each step, only the top-K states with the highest cumulative alignment scores are retained, where K denotes the beam width. After processing all n acronym characters, we select the complete alignment path with the highest score.

When $K = 1$, beam search reduces to a greedy search that only retains the highest scoring partial alignment at each step. As K increases, beam search considers a larger set of partial alignments. When K is greater than or equal to the number of valid partial alignment at every step, no state is pruned and beam search is equivalent to an exhaustive search over all valid subsequences. We study the effect of beam width K for our beam search aligner in detail in our *Ablation Studies* subsection.

4 Experiment

In this section, we empirically validate our proposed approach. First, we perform ablation studies to assess: (i) the quality of our PDA weak labels, (ii) the effectiveness of training character-level sequence labelling (CLSL) models on these weak labels, and (iii) the impact of beam width for beam search alignment. Next, we evaluate the complete end-to-end approach under different labelling strategies and heuristic baselines.

4.1 Datasets

We evaluate the generalizability of our approach by adapting three publicly available datasets for AM. These datasets originate from different domains, including corporate names (Arimond et al., 2023; Vrandečić and Krötzsch, 2014), research paper titles from arXiv (Moran, 2025), and scientific terminology (Zilio et al., 2022). We extract the acronym-long-form pairs from each dataset and ensure that all acronyms are monotonic subsequences of their long forms.

As character-level acronym-long-form annotations are costly and difficult to obtain in practice, we evaluate our approach against weakly supervised benchmarks where the training and validation sets remained unlabelled. For the test set, we construct high-quality reference labels by using the Gemini 2.0 Flash (Anil et al., 2023) LLM to generate character-level alignments between acronyms and their long forms, and manually verify and correct any misaligned mappings.

Dataset	Split	#Pos.	#Neg.	#Total
Corporate	Train	1,105	–	1,105
	Val	277	–	277
	Test	818	156	974
ArXiv Title	Train	3,129	–	3,129
	Val	769	–	769
	Test	2,469	493	2,962
Terminology	Train	20,000	–	20,000
	Val	10,000	–	10,000
	Test	9,821	1,964	11,785

Table 3: Summary of datasets used for Acronym Matching (AM). Test sets are split into positive (valid acronym-long-form pairs) and negative (synthetic invalid pairs).

To evaluate the robustness of our approach to invalid acronym-long-form-pairs, we augment the test set with negative examples. Given a long form, we randomly sample a subsequence of 3-7 characters to construct a synthetic short form. We only retain subsequences that do not coincide with the original acronym. Each test set contains approximately 20% of such negative pairs. Table 3 summarizes the evaluation datasets used in our experiments. For reproducibility, we will release all code, prompts, datasets, and experimental scripts upon publication.

4.2 Experiment Metrics

We evaluate our approach at two levels: (i) character-level alignment quality and (ii) sequence-level exact match. To evaluate character-level alignments, we only consider positive pairs and treat it as a sequence labelling task. Each character in the long form is labelled 1 if they are part of the acronym or 0 otherwise. We report precision, recall and F1 over all character labels.

To evaluate end-to-end AM, we adopt a strict sequence-level exact-match criterion. As partial matches have limited practical use in real-world applications, such as entity resolution or deduplication, an example is predicted correctly only if all acronym characters is correctly aligned to its long form. The exact match (EM) accuracy is defined in Equation 4 where N is the number of evaluation samples, $\hat{\mathbf{y}}^{(j)}$ and $\mathbf{y}^{(j)}$ are the ground-truth and predicted label sequences, and $[\cdot]$ is 1 if the condition holds true and 0 otherwise.

$$\text{EM} = \frac{1}{N} \sum_{j=1}^{N} \left[\hat{\mathbf{y}}^{(j)} = \mathbf{y}^{(j)}\right], \tag{4}$$

Method	Corporate			Arxiv Title			Terminology		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
$L_{\textbf{SHB}}$	0.7054	0.7050	0.7052	0.5039	0.5038	0.5039	0.7442	0.7436	0.7439
$L_{\textbf{SHF}}$	0.7911	0.7906	0.7908	0.6267	0.6266	0.6266	0.8042	0.8035	0.8039
$L_{\textbf{GEMINI}}$	<u>0.8748</u>	<u>0.9472</u>	<u>0.9096</u>	<u>0.9533</u>	<u>0.9764</u>	<u>0.9647</u>	<u>0.9610</u>	<u>0.9880</u>	<u>0.9743</u>
$L_{\textbf{PDA}}$	**0.9792**	**0.9791**	**0.9765**	**0.9765**	**0.9768**	**0.9767**	**0.9924**	**0.9925**	**0.9924**

Table 4: Comparison of weak label quality (Precision, Recall, F1) on training data across all three datasets. Best scores are in bold and second best are underlined.

4.3 Implementation Details and Hyperparameters

For our proposed PDA approach, we performed a small hyperparameter sweep over the decay rate $\lambda \in \{0.001, 0.01, 0.1\}$ and noticed minor variation in character-level alignment. Therefore, $\lambda = 0.01$ was selected as the default. In addition, we treat the characters " ", -, ', and _ as word delimiters.

To facilitate a fair comparison, all CLSL models are trained for up to 20 epochs using the binary cross-entropy loss with an early stop criterion of 3 epochs based on validation performance. We refer to each approach with the naming convention `<architecture>-<task>-<label>`. For instance, CANINE-SL-PDA refers to the CANINE (Clark et al., 2022) architecture, adapted for sequence labelling (SL), trained on the weak labels generated by PDA.

Model results are reported from a single training run with a fixed random seed, using the checkpoint with the best validation performance. Heuristic baselines are deterministic and are reported as single values.

4.4 Ablation Studies

We conduct three ablation studies to assess each component of our proposed approach: (i) the quality of the weak labels generated by PDA, (ii) the performance of CLSL models trained on these weak labels, and (iii) the effect of beam width K for character-level alignments.

To evaluate the effectiveness of our PDA labels, we compare it against labels generated with heuristic and LLM-based approaches. This experiment provides a rough indication on how close each weak-labelling approach is to the ground truth and their potential effectiveness as weak supervision signals. For this experiment, we only consider positive examples in the test set and assess their quality using the character-level precision, recall and F1 with respect to the reference labels. The evaluated labelling approaches are:

1. $L_{\textbf{SHB}}$: Weak labels based on greedy backward alignment with Schwartz and Hearst (2003).
2. $L_{\textbf{SHF}}$: Weak labels based on greedy forward alignment with Schwartz and Hearst (2003).
3. $L_{\textbf{GEMINI}}$: Weak labels based on character-level alignments from the Gemini 2.0 Flash (Anil et al., 2023) LLM.
4. $L_{\textbf{PDA}}$: Weak labels based on our proposed PDA approach.

Table 4 summarizes our results where higher scores indicate stronger alignment to the ground truth. Across all three datasets, our proposed L_{PDA} achieves high precision, recall and F1, exceeding 97%. In addition, L_{PDA} consistently outperforms heuristic-based weak labels (L_{SHB} and L_{SHF}), as well as LLM-generated labels, L_{GEMINI}. These results demonstrate that PDA generated labels are strongly aligned with the ground truth, highlighting their efficacy as weak supervision signals.

Next, we evaluate the effectiveness of our CLSL models trained on L_{PDA}. We consider two encoder architectures: (i) CANINE (Clark et al., 2022), a character-level encoder, and (ii) ByT5 (Xue et al., 2022), a byte-level encoder. We evaluate our approach on the positive examples in the test set and report the character-level precision, recall and F1. Table 5 summarizes our findings. For both architectures, PDA-based approaches outperform heuristic-based baselines (SHB and SHF) in terms of precision and overall F1. In addition, PDA-based models outperform or achieve similar F1 scores to variants trained on LLM-generated labels, with a difference less than 0.005. These results suggest that CLSL models trained on PDA generated labels achieve performance comparable to those trained on labels generated by Gemini, while requiring substantially less computational resources to generate labels for weak supervision.

Lastly, to determine an appropriate beam width K for our beam-search aligner, we compare beam

Method	Corporate			ArXiv Title			Terminology		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
CANINE-SL-GEMINI	0.8369	0.6936	0.7585	0.9504	0.5581	0.7032	0.9471	**0.8039**	**0.8696**
CANINE-SL-SHB	0.6303	0.7378	0.6798	0.4594	0.4529	0.4561	0.7158	0.6441	0.6780
CANINE-SL-SHF	0.6932	**0.7670**	0.7283	0.6456	**0.6919**	0.6680	0.6698	0.6744	0.6721
CANINE-SL-PDA	**0.8504**	0.6872	**0.7601**	**0.9291**	0.6242	**0.7467**	**0.9480**	0.7988	0.8670
BYT5-SL-GEMINI	0.8418	0.7029	0.7661	0.8787	0.6152	0.7237	**0.9518**	**0.8008**	**0.8698**
BYT5-SL-SHB	0.6679	0.5323	0.5924	0.5132	0.4374	0.4723	0.7280	0.7997	0.7622
BYT5-SL-SHF	0.6962	0.5484	0.6135	0.4837	0.4559	0.4694	0.7209	0.6563	0.6871
BYT5-SL-PDA	**0.8517**	**0.7057**	**0.7719**	**0.8932**	**0.6332**	**0.7410**	0.9344	0.8053	0.8651

Table 5: Character-level precision, recall, and F1 for CANINE and ByT5-based sequence labelling models trained with different labelling strategies. Best scores for each architecture are in bold and second best are underlined.

Method	Corporate	Arxiv T.	Terminology
Baseline	0.8604	0.9369	0.9620
BS $K = 1$	0.7977	0.9264	0.9191
BS $K = 2$	0.8552	0.9362	0.9605
BS $K = 3$	0.8593	0.9375	0.9619
BS $K = 5$	0.8593	0.9379	0.9620
BS $K = 10$	0.8604	0.9369	0.9620
BS $K = 100$	0.8604	0.9369	0.9620

Table 6: Sequence-level exact match (EM) accuracy for the baseline (exhaustive alignment) and beam search with varying beam widths K.

Pos	Char	Score	$K = 1$	$K = 2$	$K = 3$
1	**M**	0.8050			**M**
2	**T**	0.1408			**T**
3	**S**	0.032			**S**
4		0.002			
5	**A**	0.8865			**A**
6	S	0.1062			
7	S	0.1062			
8	O	0.0015			
9	C	0.0014			
10	I	0.0019			
11	A	0.0015			
12	T	0.0076			
13	E	0.0012			
14	D	0.0024			
15		0.0006			
16	**M**	0.9309	**M**	**M**	**M**
17	A	0.1442			
	...				

Table 7: Example of the acronym MTSAM mapped to its long form ***MTS ASSOCIATED MARKETS*** for different beam widths K. The ground truth and the highest scoring alignment for each approach are denoted in bold.

width of different sizes $K = \{1, 2, 3, 5, 10, 100\}$ against an exhaustive alignment baseline. This baseline iterates through all valid subsequences and selects the highest scoring one. For consistency, we use CANINE-SL-PDA to generate the character-level scores. In this experiment we consider both positive and negative examples in the test set and evaluate each configuration using the EM accuracy in Equation 4. The results are summarized in Table 6. We observe that a beam width of $K = 3$ achieves comparable EM accuracy to the exhaustive baseline with a difference of less than 0.0012. To illustrate how beam width affects alignment quality, Table 7 presents a case study for the acronym MTSAM and the long form *MTS ASSOCIATED MARKETS*. When $K = 3$, the correct alignment is recovered, whereas when $K < 3$, the algorithm prioritize the higher scoring character 'M' at index 16, resulting in an incomplete alignment. Additionally, when $K = 3$, beam search avoids exploring subsequence containing low-scoring characters such as character 'S' at position 7. For the remaining experiments, we set $K = 3$.

4.5 End-to-End Evaluation

In this subsection, we evaluate our end-to-end AM framework which comprises: (i) a CLSL model trained on PDA labels and (ii) a beam-search decoder for monotonic alignment. In addition to the variants detailed in the ablation studies, we also include six additional benchmarks:

1. **SHB**: Greedy backward alignment with Schwartz and Hearst (2003).
2. **SHF**: Greedy forward alignment with Schwartz and Hearst (2003).
3. **IM**: Rule-based alignment that matches acronym characters to first character of each word (initials) in the long form.
4. **SHB-IM**: Combination of SHB and IM.
5. **SHF-IM**: Combination of SHF and IM.

Method	Corporate	Arxiv T.	Term.
CANINE-AM-GEM	0.8450	0.9301	0.9691
CANINE-AM-SHB	0.6879	0.3265	0.5963
CANINE-AM-SHF	0.8316	0.7691	0.5825
CANINE-AM-PDA	0.8593	**0.9375**	0.9619
BYT5-AM-GEM	0.8645	0.9072	**0.9702**
BYT5-AM-SHB	0.6745	0.4733	0.8808
BYT5-AM-SHF	0.7136	0.4666	0.5789
BYT5-AM-PDA	**0.8676**	0.9200	0.9625
SHB	0.3101	0.1165	0.3874
SHF	0.3922	0.1745	0.4417
IM	0.4661	0.2205	0.8048
SHB-IM	0.4908	0.1553	0.7230
SHF-IM	0.5411	0.2151	0.7371
PDA	0.8070	0.8173	0.8392

Table 8: End-to-end evaluation: EM-All accuracy of acronym matching (AM) across all three evaluation datasets: Corporate, Arxiv Title (Arxiv T.) and Terminology (Term.). Best scores are in bold and second best are underlined.

Method	Alignment
CANINE-AM-GEM	**C**ROWTHORNE **O**LD **A**GE **T**O TEEN **S**OCIETY
CANINE-AM-PDA	**C**ROWTHORNE **O**LD **A**GE TO **T**EEN **S**OCIETY
PDA	**C**ROWTHORNE **O**LD **A**GE **T**O TEEN **S**OCIETY
SHB	**C**ROWTHORNE **O**LD **A**GE TO **T**EEN **S**OCIETY
CANINE-AM-GEM	**G**LOBAL **R**ISK **A**ND ASSET **M**ANAGEMENT **CO**MPANY
CANINE-AM-PDA	**G**LOBAL **R**ISK AND **A**SSET **M**ANAGEMENT **CO**MPANY
PDA	**G**LOBAL **R**ISK **A**ND ASSET **M**ANAGEMENT **CO**MPANY
SHB	**G**LOBAL **R**ISK AND **A**SSET MANAGE**M**ENT **CO**MPANY

Table 9: Acronym-long-form alignments for (i) COATS, *CROWTHORNE OLD AGE TO TEEN SOCIETY* and (ii) GRAMCO, *GLOBAL RISK AND ASSET MANAGEMENT COMPANY*.

6. **PDA**: Our proposed PDA approach (used to generate weak labels).

The baselines SBH-IM and SHF-IM correspond to our adaptation of the rule-based approach proposed by Veyseh et al. (2021). In addition, we compare our proposed end-to-end framework against PDA. This allows us to assess the effectiveness of combining a CLSL model trained on PDA labels with a monotonic beam search aligner against the underlying heuristics used to generate PDA labels. In this experiment, we consider both positive and negative examples in the test set and evaluate the performance using EM accuracy. The results are presented in Table 8. For the corporate and Arxiv title datasets, our proposed frameworks achieve the highest performance, outperforming its LLM-based counterpart. On the terminology dataset, BYT5-AM-PDA achieves EM accuracy on par with BYT5-AM-GEMINI with a difference of 0.0077. In addition, our end-to-end frameworks consistently outperform heuristic based approaches for all three datasets.

To better understand the mappings produced by each approach, we refer to the case study (i) COATS, *CROWTHORNE OLD AGE TO TEEN SOCIETY*, in Table 9. For the character 'T', PDA selects the first word with the corresponding initial ***TO*** whereas CAININE-AM-PDA aligns 'T' to the more informative word ***TEEN***. Closer inspection on the CLSL scores show that the model place higher emphasis on informative words than stopwords: ***TO*** has a score of 0.2767 whereas ***TEEN*** has a score of 0.8865. This allows the beam-search aligner to correctly map 'T' to the more informative word. Similarly, this can also be observed in our second example (ii) GRAMCO, *GLOBAL RISK AND ASSET MANAGEMENT COMPANY*. These findings suggests that, although the CLSL model is trained primarily on position-biased labels, it is capable of distinguishing characters in informative words from those in stopwords. This allows our approach to favour informative words over stopwords when multiple alignments exist.

5 Conclusion

In this work, we formulate acronym matching as a monotonic character subsequence alignment problem and propose a weakly supervised approach that effectively aligns acronym characters to their corresponding long form characters. Extensively evaluated on three AM datasets, our proposed approach proved effective, outperforming heuristic baselines and achieving comparable results to variants trained on LLM-generated labels while requiring substantially less compute. This demonstrates the effectiveness of our approach in resource constrained scenarios where character-level annotations and access to LLMs are limited.

6 Limitations

Despite these advantages, our current formulation is restricted to monotonic subsequence alignment, which requires all characters in the acronym to appear in the long form. Therefore, this approach

is unable to directly handle semantically equivalent but textually dissimilar mappings, such as "&" and "and". Future work will focus on extending this framework to handle semantic variation and explore multilingual and non-Latin mappings.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, and 260 others. 2023. Gpt-4 technical report.

Izhar Ali, Million Haileyesus, Serhiy Hnatyshyn, Jan-Lucas Ott, and Vasil Hnatyshin. 2024. Automated extraction of acronym-expansion pairs from scientific papers.

Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, and Oriol Vinyals. 2023. Gemini: A family of highly capable multimodal models.

Alexander Arimond, Mauro Molteni, Dominik Jany, Zornitsa Manolova, Damian Borth, and Andreas G. F. Hoepner. 2023. Transformer-based entity legal form classification. *Preprint*, arXiv:2310.12766.

Keno Bressem, Michalis Papaioannou, Paul Grundmann, Florian Borchertd, Lisa Adams, Leonhard Liu, Felix Busch, Lina Xu, Jan Loyen, Stefan Niehues, Moritz Augustin, Lennart Grosser, Marcus Makowski, Hugo Aerts, and Alexander Löser. 2023. medbert.de: A comprehensive german bert model for the medical domain. *Expert Systems with Applications*, 237:121598.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language understanding. *Transactions of the Association for Computational Linguistics*.

Nicholas Egan and John Bohannon. 2020. Primer ai's systems for acronym identification and disambiguation. *ArXiv*, abs/2012.08013.

Bikash Gyawali, Lucas Anastasiou, and Petr Knoth. 2020. Deduplication of scholarly documents using locality sensitive hashing and word embeddings. In *International Conference on Language Resources and Evaluation*.

Amila Kugic, Stefan Schulz, and Markus Kreuzthaler. 2025. Embedding-based acronym disambiguation supported by large language models in german clinical narratives.

Sean Moran. 2025. Acronymgen-titles: Acronym–title pairs from arxiv. https://huggingface.co/datasets/sjmoran/arxiv-acronym-gen. Accessed: 2025-11-28.

Jose Ricardo Oliveira and Adriano Galindo Leal. 2025. Enhancing anti-money laundering protocols: Employing machine learning to minimise false positives and improve operational cost efficiency. In *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence*, ICAAI '24, page 8–13, New York, NY, USA. Association for Computing Machinery.

Berkan Oztas, Deniz Cetinkaya, Festus Adedoyin, Marcin Budka, Gokhan Aksu, and Huseyin Dogan. 2024. Transaction monitoring in anti-money laundering: A qualitative analysis and points of view from industry. *Future Generation Computer Systems*, 159:161–171.

Harsh Patel, Dominique Boucher, Emad Fallahzadeh, Ahmed E. Hassan, and Bram Adams. 2025. A state-of-the-practice release-readiness checklist for generative ai-based software products: A gray literature survey. *IEEE Software*, 42(1):74–83.

Ariel Schwartz and Marti Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 4:451–62.

Amir Veyseh, Franck Dernoncourt, Walter Chang, and Thien Nguyen. 2021. Maddog: A web-based system for acronym identification and disambiguation. 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, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3285–3301. International Committee on Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Zhi Wen, Xing Han Lu, and Siva Reddy. 2020. MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 130–135, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Wenli Yang, Lilian Some, Michael Bain, and Byeong Kang. 2025. A comprehensive survey on integrating large language models with knowledge-based methods. *Knowledge-Based Systems*, 318:113503.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA.

Leonardo Zilio, Hadeel Saadany, Prashant Sharma, Diptesh Kanojia, and Constantin Orăsan. 2022. PLOD: An abbreviation detection dataset for scientific documents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 680–688, Marseille, France. European Language Resources Association.