Dynamically Acquiring Text Content to Enable the Classification of Lesser-known Entities for Real-world Tasks

Fahmida Alam, Ellen Riloff
University of Arizona, Tucson, AZ, USA
{fahmidaalam, riloff}@arizona.edu

Abstract

Existing Natural Language Processing (NLP) resources often lack the task-specific information required for real-world problems and provide limited coverage of lesser-known or newly introduced entities. For example, business organizations and health care providers may need to be classified into a variety of different taxonomic schemes for specific application tasks. Our goal is to enable domain experts to easily create a task-specific classifier for entities by providing only entity names and gold labels as training data. Our framework then dynamically acquires descriptive text about each entity, which is subsequently used as the basis for producing a text-based classifier. We propose a novel text acquisition method that leverages both web and large language models (LLMs). We evaluate our proposed framework on two classification problems in distinct domains: (i) classifying organizations into Standard Industrial Classification (SIC) Codes[1], which categorize organizations based on their business activities; and (ii) classifying healthcare providers into healthcare provider taxonomy codes[2], which represent a provider's medical specialty and area of practice. Our best-performing model achieved macro-averaged F1-scores of 82.3% and 72.9% on the SIC code and healthcare taxonomy code classification tasks, respectively.

Keywords: automatic text acquisition, web and LLM-based retrieval, task-specific classification

1. Introduction

Many real-world applications require knowledge about named entities, including organizations, people, and places. To address this need, researchers have developed structured data resources, such as knowledge bases and knowledge graphs, that compile information about a wide range of named entities. DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), and Yago (Suchanek et al., 2007) are examples of widely used structured knowledge resources. These resources typically capture attributes of general interest, such as the location and size of organizations or the professions of individuals, making them broadly useful.

However, many real-world applications require entity-specific knowledge that is not present in existing resources. For instance, applications may need to classify organizations by their business activities, revenue sources, client types, or ownership structures. Furthermore, categorization schemes can vary significantly across applications. For example, business activities could be characterized by a small number of high-level categories or broken down into a large number of fine-grained categories that are relevant to the application domain.

Our goal is to develop a method that can rapidly acquire new types of information about named entities given only their names as input, without the need for an explicit text corpus or structured knowledge resource to enable the learning. We introduce a framework that acquires descriptive text for entities given only their names as input and subsequently trains classifiers using the acquired text along with the corresponding gold labels. For text acquisition, the framework integrates web retrieval and LLM-based generation to automatically produce descriptive text for each entity. The framework is adaptable, scalable, and requires minimal human supervision, supporting the rapid development of entity classification systems across diverse tasks and domains. To demonstrate its generalizability, we applied the framework to two classification tasks from entirely different domains, and the results highlight its effectiveness and robustness. All in all, our contributions are:

1. We propose a generalizable framework that takes only the entity names and their corresponding gold labels as input. It handles the entire process through a novel text acquisition method that leverages web retrieval and LLM-based generation to produce descriptive text for classifier training. This approach eliminates dependence on pre-compiled datasets containing entity descriptions, which represents the key novelty of our work.

2. We evaluate our framework on two different types of real-world classification tasks: (i) clas-

[1]The U.S. government established Standard Industrial Classification (SIC) codes to classify organizations based on the activities they serve and operate within.

[2]The Health Care Provider Taxonomy code set, maintained by the National Uniform Claim Committee (NUCC), classifies health care providers by grouping, classification, and area of specialization, such as Cardiology, Obstetrics & Gynecology, or Family Medicine.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10815–10825
11–16 May 2026.

sifying organizations into Standard Industrial Classification (SIC) codes and (ii) classifying healthcare providers into healthcare provider taxonomy codes.

3. We constructed two benchmark datasets using our framework in two distinct domains: (i) industry and (ii) healthcare, to demonstrate its effectiveness and generalizability. Evaluation results indicate that our framework achieves robust performance across domains. We release both datasets on GitHub[3] to facilitate future research in automated knowledge acquisition, text-based classification, and entity classification, while following responsible data-sharing practices.

2. Related Work

Named entity recognition and entity classification have been extensively studied in NLP, but have traditionally focused on labeling entity mentions in a document or text fragment (e.g., (Nadeau and Sekine, 2007; Ling and Weld, 2012; Shen et al., 2012; Yaghoobzadeh and Schütze, 2015; Li et al., 2023)). In contrast, our research aims to acquire knowledge about real-world entities irrespective of any specific mention or document.

Our research shares the same high-level goal as Knowledge Base Population (KBP), which extracts information to populate structured knowledge bases, such as Wikidata, DBpedia, or TAC-KBP (Ji and Grishman, 2011; Surdeanu, 2013). Prior work has utilized weak or distant supervision by leveraging structured knowledge resources like Wikipedia or Freebase to acquire training data (e.g., (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011)). However, our approach does not use any data from structured knowledge resources or external corpora. Our goal is to acquire knowledge that is not currently available in existing resources, to augment them or populate new ones.

Although our method utilizes retrieved text, it differs fundamentally from Retrieval-Augmented Generation (RAG) approaches, which typically retrieve documents from a static, indexed corpus, such as Wikipedia, The Pile (Gao et al., 2020), or similar sources, to guide a generative model at inference time (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021). In contrast, our approach performs web search to proactively retrieve context and leverages multiple LLMs to generate task-specific text. These sources, individually and in combination, are then used to construct training data for multiple classifiers trained under a supervised learning paradigm. Some prior work has integrated search engine retrieval into language models (Karpukhin et al., 2020; Komeili et al., 2021), but such approaches are primarily designed for document ranking or generation tasks, rather than training a classifier to assign labels based on retrieved content. To the best of our knowledge, no prior work has combined web-retrieved and LLM-generated text as complementary sources to construct training data for supervised classification.

Our work was inspired by the research of (Jiang et al., 2023), which used google-retrieved text to classify organizations for knowledge-graph population in the food systems domain. However, our framework goes beyond web retrieval by also leveraging multiple LLMs for task-specific text generation, exploring whether LLMs can reduce dependence on existing structured knowledge sources for various NLP tasks. Moreover, while their experiments were limited to BERT-based models, we evaluate our framework using three additional language models, RoBERTa, Longformer, and GPT-4o mini, to assess performance across different model families. Furthermore, we experimented with confidence-based thresholds to produce high-precision predictions, enabling the automatic annotation of new training data with minimal human intervention.

3. Task Definition and Dataset

Although our experiments focus on the following tasks, the proposed framework is task-agnostic and can be adapted to a wide range of entity-centric categorization and knowledge acquisition tasks.

3.1. SIC Code

Task Definition In the SIC code classification task, organizations are categorized by their Standard Industrial Classification (SIC) codes, which describe their primary business activities and are assigned by the U.S. government.[4] SIC codes are widely used for economic analyses and food supply chain studies (Jiang et al., 2023). These codes follow a hierarchical structure: a four-digit code specifies an industry nested within broader categories. For example, the code 0116 represents the Soybeans industry under Cash Grains, within Agricultural Production Crops. This hierarchy enables analysis at varying levels of granularity by grouping organizations by the first one to four digits.

Dataset As a starting point, we adopt the dataset introduced by (Jiang et al., 2023), originally de-

[3] https://github.com/alamfahmida/dynamic-text-acquisition-entity-classification

[4] These codes are analogous to North American Industry Classification System (NAICS) codes used across North America.

veloped for food system research, and enrich it with task-relevant texts harvested using our framework. The organizations in this dataset primarily consist of lesser-known organizations rather than widely recognized corporations like Google or IBM. Most of them, such as Multi-Corp International Inc., Fonon Corp., and JMXI Inc., are not commonly mentioned in general resources.[5] Following their setup, we use the first two digits of the SIC codes as category labels to reduce sparsity and capture each organization's broad business activities. The dataset contains 5,400 organizations labeled with two-digit SIC codes across 27 categories (see Table 1). The dataset is partitioned into 2,700 training, 900 development, and 1,800 test instances.

SIC Code Categories
Metal Mining; Oil and Gas Extraction; Food and Kindred Products; Printing, Publishing and Allied Industries; Chemicals and Allied Products; Fabricated Metal Products; Industrial and Commercial Machinery; Electronic; Transportation Equipment; Measuring, Photographic, Medical, Communications; Electric, Gas and Sanitary Services; Wholesale Trade - Durable Goods; Wholesale Trade - Nondurable Goods; Eating and Drinking Places; Miscellaneous Retail; Depository Institutions; Nondepository Credit Institutions; Security; Insurance Carriers; Real Estate; Holding and Other Investment Offices; Hotels, Rooming Houses, Camps; Business Services; Amusement and Recreation Services; Health Services; Engineering, Accounting, Research
Healthcare Taxonomy Code Categories
Allopathic & Osteopathic Physicians; Behavioral Health and Social Service Providers; Chiropractic Providers; Dental Providers; Dietary and Nutritional Service Providers; Emergency Medical Service Providers; Eye and Vision Service Providers; Nursing Service Providers; Nursing Service Related Providers; Other Service Providers; Pharmacy Service Providers; Physician Assistants and Advanced Practice Nursing Providers; Podiatric Medicine and Surgery Service Providers; Respiratory, Developmental, Rehabilitative and Restorative Service Providers; Speech, Language and Hearing Service Providers; Student, Health Care; Technologists, Technicians, and Other Technical Service Providers

Table 1: Categories used in the SIC code (27 categories) and healthcare provider taxonomy code (17 categories) classification tasks.

3.2. Healthcare Provider Taxonomy Code

Task Definition The healthcare provider taxonomy classification task involves assigning healthcare professionals to their corresponding taxonomy codes, which define their medical specialty and area of practice. Each taxonomy code is a ten-character alphanumeric identifier from the Health Care Provider Taxonomy Code Set maintained by the National Uniform Claim Committee (NUCC)[6]. The code follows a three-level hierarchy: (i) Provider Grouping, a broad category such as Allopathic and Osteopathic Physicians; (ii) Classification, a discipline within the group, e.g., Internal Medicine; and (iii) Specialization, a focused area like Cardiovascular Disease. For example, `207RC0000X` represents the grouping `Allopathic and Osteopathic Physicians`, classification `Internal Medicine`, and specialization `Cardiovascular Disease`. Thus, by identifying a provider's taxonomy code, one can infer these specific details about their professional domain.

Dataset To start, we took provider names and their corresponding gold categories from the National Plan and Provider Enumeration System (NPPES)[7] . As mentioned earlier, the 10-digit taxonomy follows a three-level hierarchy, but the NUCC documentation does not specify digit boundaries for each level. Therefore, we use the full 10-digit taxonomy codes. The code set covers individuals, groups, and non-individual entities. Since we focus on healthcare providers' taxonomy code, we target only the `individual category`, which contains information exclusively about individual healthcare providers. There are a total of 17 fixed categories (see Table 1), each containing multiple taxonomy codes. For instance, the category *Allopathic & Osteopathic Physicians* includes codes such as `207K00000X` and `2081S0010X`, etc.

To construct a balanced subset across the taxonomy hierarchy, we focused on 17 individual categories and selected one taxonomy code from each category that had at least 200 individual provider instances in the NPPES database. Then we randomly sampled 200 individual providers, resulting in a total of 3,400 instances. We then split the dataset into training, development, and test sets using the same ratio as in our SIC classification task, yielding 1,700 training, 1,140 test, and 560 development instances.

[5]The dataset released on GitHub includes the complete list of organizations.

[6]https://nucc.org/

[7]The National Plan and Provider Enumeration System (NPPES) is a public registry maintained by the Centers for Medicare & Medicaid Services (CMS). It includes information on U.S. healthcare providers, their specialty taxonomy codes, and National Provider Identifiers (NPIs). Links: NPPES, https://www.cms.gov/.

4. Proposed Framework

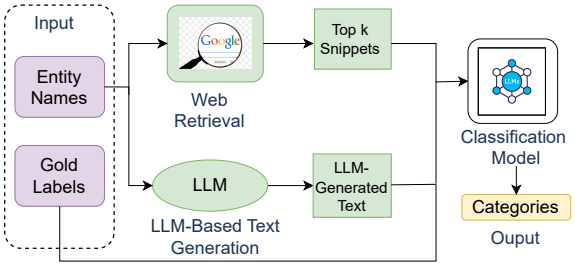


Figure 1: Overview of the proposed architecture. The input consists of entity names and their corresponding gold labels. The framework employs two components for text acquisition: (i) a web retrieval module that retrieves top-*k* snippets, and (ii) an LLM-based module that generates task-specific descriptive text. The retrieved and generated texts, along with their gold labels, are then used to train a classification model that predicts the task-specific category for a given entity.

We propose a framework that takes an entity as input and autonomously handles the entire process, from text acquisition to classifier training. For training purposes, the framework requires gold labels for the training set, which are not used during text acquisition. Figure 1 illustrates the components and workflow of our architecture. The framework consists of two main steps: (1) TEXT ACQUISITION and (2) CLASSIFICATION MODEL TRAINING.

4.1. Text Acquisition

The key idea behind our approach is that entities can be categorized using text descriptions that are *automatically* acquired for the specific application task, without relying on any pre-compiled text resources. We combine two approaches to obtain task-relevant text for each entity: (1) retrieving text using a search engine and (2) generating text with large language models. For the SIC code classification task, "task-relevant" text refers to descriptions of an organization's business activities, whereas for the healthcare taxonomy code classification task, it refers to information about a provider's medical specialty, professional focus, and area of practice.

4.1.1. Google Snippets

We used the SerpAPI[8] interface to programmatically query the Google Search Engine[9] for each entity name (organization or healthcare provider), without quotes. For each query, we considered the top 10 search results. In our experiments, we used text snippets, which are small blocks of text that appear underneath a link to a webpage in a search engine results page. These snippets are typically around 100–200 characters long and provide a short description of the webpage content. We obtained the snippets from the top 10 retrieved results and concatenated them into a single text block. We refer to this aggregated text as *GSnip*, which serves as semantically rich content capturing how the entity is described across multiple sources. Figure 2 shows the *GSnip* text acquired for the organization *Gold Hills Mining, Ltd.*.

GSnip

Gold Hills Mining, Ltd. develops gold, silver, and copper mines around the world, with strong emphasis on Latin America and on Brazil in particular. Gold Hills Mining, Ltd is primarely in the business of gold & silver ores. For financial reporting, their fiscal year ends on June 30th. This page includes all ... Goldhills strategy is to generate or acquire early stage precious metals exploration opportunities and advance them through direct exploration by our ... A high-level overview of Gold Hills Mining, Ltd. (GHML) stock. Stay up to date on the latest stock price, chart, news, analysis, fundamentals, ... GOLD HILLS MINING LT. GHML. Delayed OTC Markets - 01:30 2016-03-31 pm EDT. 0.4500 USD, -10.00%. Prev. 0.50000000. Open, 0.45000000. High, 0.45000000. Gold Hills Mining, Ltd., Massive Dynamics, Inc., Medisafe 1 Technologies Corp., and MDU. Communications International, Inc. The revocation ... Gold Hills Mining Ltd ... Advertiser Disclosure: TD Ameritrade, Inc. and Accretive Capital LLC are separate, unaffiliated companies and are not responsible ... Financial Summary for Gold Hills Mining Ltd (GHML) showing last 5-quarter or 5-year trends for Income Statement, Balance Sheet, and Cash Flow. Real-time Price Updates for Gold Hills Mining Ltd (GHML), along with buy or sell indicators, analysis, charts, historical performance, news and more. Title. Gold Hill Mining and Milling Company: placer mine. Date. [1890-1900]. Call Number. BHS 219-3-39. Contents. 10 photographs (3 views). Description.

Figure 2: Example *GSnip* for organization *Gold Hills Mining, Ltd.*.

4.1.2. LLM-Generated Summaries

We also generated task-specific summaries using two large language models (LLMs): GPT-4O mini[10] (OpenAI, 2024) and LLAMA 3.1–8B INSTRUCT[11] (MetaAI, 2024). These summaries aim to concisely capture the key characteristics of each entity. By comparing summaries generated by different language models, we assess the consistency and semantic quality of LLM-generated text descriptions. Due to architectural and alignment differences between models, we adopted distinct prompting strategies.

GPT-generated Summaries For the SIC code task, we prompted the model with: "`Summarize the main business activities, services, vision, and mission of [ORG_NAME].`". For the healthcare taxonomy code task, the prompt was: "`Summarize the healthcare specialization, scope of practice, and typical services provided by [Provider_NAME]. The summary should describe the clinician's professional type and main field of practice, following standard U.S. healthcare taxonomy conventions.`". We refer to the resulting summary as *GPTSum*. The aggregated Google Snippets (GSnip) are typically around 250–300 words, so we set the model's max-

[8] https://serpapi.com/

[9] https://programmablesearchengine.google.com/about/

[10] Specifically, we used `gpt-4o-mini-2024-07-18`.

[11] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

imum token limit to produce summaries of comparable length. Figure 3 shows an example of *GPTSum* for *Gold Hills Mining, Ltd.*.


GOLD HILLS MINING, LTD. is a company focused on the exploration, development, and production of gold mining properties.

Main Business Activities:
- **Exploration:** Identifying and evaluating potential gold mining sites.
- **Development:** Preparing identified sites for mining operations through infrastructure development and resource estimation.
- **Production:** Extracting gold from operational mines and processing it for sale.

Services:
- Geological surveys and assessments.
- Mining operation management.
- Gold extraction and processing.
- Environmental management and sustainability consulting.

Vision:
To become a leading and sustainable gold producer, recognized globally for its innovative and responsible mining practices, contributing to economic growth and environmental stewardship.

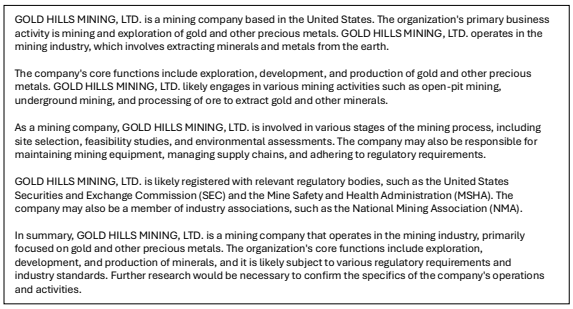
Mission:
To deliver high-quality gold products while ensuring safety, environmental sustainability, and community development. The company aims to achieve this through efficient operations, ethical practices, and a commitment to social responsibility.

Figure 3: Example of *GPTSum* for *Gold Hills Mining, Ltd.*.

LLaMA-generated Summaries Being smaller and more sensitive to prompt specificity, LLAMA 3.1–8B INSTRUCT benefited from more explicit instruction designed to constrain text generations to factual and verifiable content. This helped reduce hallucination and ensured consistency in structure and coverage with GPT-generated summaries. For the SIC code task, the prompt was: "`You are an assistant writing a factual summary about an organization based on its name. Given the [ORG_NAME], your goal is to identify and describe the organization's main business activities, core functions, and the industry it operates in. Use only publicly verifiable information. The description should be informative, objective, and around 250–300 words. Do not add any assumptions or speculative content.`". For the healthcare taxonomy task, the prompt was: "`You are a research assistant writing a factual summary about a healthcare provider's specialty. Given the [PROVIDER_NAME], your goal is to identify and describe their medical specialty, professional focus, qualifications, and the healthcare sector they operate in. Use only publicly verifiable information. The description should be informative, objective, and around 250–300 words. Do not add any assumptions or speculative content.`". The resulting summary is referred to as *LLaMASum*. An example of *LLaMASum* is illustrated in the Figure 4 for the entity *Gold Hills Mining, Ltd.*.

4.2. Classification Model Training

We trained classification models by fine-tuning both encoder-based and generative language models. Each model was evaluated using five types of texts:


GOLD HILLS MINING, LTD. is a mining company based in the United States. The organization's primary business activity is mining and exploration of gold and other precious metals. GOLD HILLS MINING, LTD. operates in the mining industry, which involves extracting minerals and metals from the earth.

The company's core functions include exploration, development, and production of gold and other precious metals. GOLD HILLS MINING, LTD. likely engages in various mining activities such as open-pit mining, underground mining, and processing of ore to extract gold and other minerals.

As a mining company, GOLD HILLS MINING, LTD. is involved in various stages of the mining process, including site selection, feasibility studies, and environmental assessments. The company may also be responsible for maintaining mining equipment, managing supply chains, and adhering to regulatory requirements.

GOLD HILLS MINING, LTD. is likely registered with relevant regulatory bodies, such as the United States Securities and Exchange Commission (SEC) and the Mine Safety and Health Administration (MSHA). The company may also be a member of industry associations, such as the National Mining Association (NMA).

In summary, GOLD HILLS MINING, LTD. is a mining company that operates in the mining industry, primarily focused on gold and other precious metals. The organization's core functions include exploration, development, and production of minerals, and it is likely subject to various regulatory requirements and industry standards. Further research would be necessary to confirm the specifics of the company's operations and activities.

Figure 4: Example of *LLaMASum* for *Gold Hills Mining, Ltd.*.

(1) *GSnip*, (2) *GPTSum*, (3) *LLaMASum*, (4) *GSnip* + *GPTSum*, and (5) *GSnip* + *LLaMASum*.

4.2.1. Encoder-based Language Models

We fine-tuned three encoder-based language models: (1) BERT (Devlin et al., 2018), (2) RoBERTa (Liu et al., 2019), and (3) Longformer (Beltagy et al., 2020). Each model was trained on our training set using identical hyperparameters, which are described below. For each entity, the input consists of the entity name concatenated with its corresponding text content, and its gold category label. Hyperparameter values were selected on the basis of performance in the development set.

BERT We fine-tuned the BERT model (`bert-base-uncased`) to create a classifier for our tasks. Input texts were tokenized using BERT's tokenizer with a maximum sequence length of 512 tokens. The final hidden state of the [CLS] token was fed into a linear classification layer to produce the predicted code.

RoBERTa We also fine-tuned the RoBERTa model (`roberta-base`), which shares the same transformer architecture as BERT but is pre-trained on a larger corpus using dynamic masking and without the Next Sentence Prediction objective. Input texts were tokenized using the RoBERTa tokenizer with a maximum sequence length of 512 tokens. As with BERT, classification was performed using the final hidden state of the first token (the [CLS] equivalent in RoBERTa).

Longformer We fine-tuned Longformer (`allenai/longformer-base-4096`) to assess whether its sparse attention mechanism offers any advantage in classification tasks. Inputs were tokenized using the Longformer tokenizer with a maximum sequence length of 1,024 tokens. The final hidden state of the [CLS] token configured with global attention was used as input to a linear classification layer.

Hyperparameters for BERT, RoBERTa, and Longformer We fine-tuned all three models using the same hyperparameters: 3 training epochs, the AdamW optimizer, a learning rate of 5e-5, 500 warmup steps, and a weight decay of 0.01. We used a training batch size of 8 and an evaluation batch size of 16, and evaluated performance at the end of each epoch.

4.2.2. Generative Language Model

GPT-4o mini We also fine-tuned GPT-4o mini (`gpt-4o-mini-2024-07-18`), a lightweight and cost-effective generative language model. We prepared training, dev, and test sets in an OpenAI-compatible chat format. The training and dev sets follow the same structure: each instance includes a system instruction and a user prompt consisting of an entity name concatenated with its acquired text description, paired with its gold label. In the test set, the gold label is omitted. Examples of the training and dev set formats are shown in Figure 7, and the test format is shown in Figure 8 in Appendix A. We followed the same format for the healthcare taxonomy classification task. The model was fine-tuned using supervised learning with default hyperparameters (`batch_size`, `n_epochs`, and `learning_rate_multiplier` set to `"auto"`).

5. Experiments and Results

In this section, we report the results on both tasks, evaluated using macro-averaged precision (P), recall (R), and F1-score.

5.1. Prompting Baselines

As a baseline, we experimented with prompting to determine whether state-of-the-art LLMs can effectively assign SIC categories to organizations and taxonomy codes to healthcare providers, which is important to assess whether the model has encountered such codes in its pre-training data. We used the GPT-4o mini[12] language model for these experiments because of its strong performance-efficiency trade-off and suitability for large-scale prompting experiments at low cost. We experimented with two prompting settings, one without additional context and another incorporating task-relevant text as context. For the no-context setting, the prompt was: "`You are a classification assistant for [TASK_NAME]. Given the [ENTITY_NAME] below, predict the [CODE_TYPE] that best represents its [CATEGORY_DESCRIPTION]. Choose ONLY one code from the provided options: [CODE_LIST]. Return ONLY the code. Do not include explanations or extra text.`"

[12]We used `gpt-4o-mini-2024-07-18` for these experiments.

GPT-4o mini	SIC Code			Healthcare Taxonomy		
	P	R	F1	P	R	F1
Prompting	0.572	0.453	0.464	0.039	0.052	0.021
Prompting w/						
LLaMASum	0.536	0.466	0.464	0.023	0.058	0.010
GPTSum	0.565	0.496	0.497	0.111	0.055	0.046
GSnip	**0.653**	0.611	**0.601**	**0.289**	**0.272**	**0.256**

Table 2: Prompting performance of GPT-4o mini across two classification tasks.

For the with-context setting, we used the same prompt, except that additional context was included along with the entity name. As context, we used three types of text: *GSnip*, *GPTSum*, and *LLaMASum*. This experiment evaluates whether providing additional text as context can improve the LLM's prompting performance. We used the same prompt structure for both the SIC code and healthcare provider taxonomy classification tasks, substituting the `[TASK_NAME]`, `[ENTITY_NAME]`, `[DESCRIPTION]`, `CATEGORY_DESCRIPTION`, and `[CODE_LIST]` accordingly.

5.1.1. Baseline Results

Table 2 shows that prompting GPT-4o mini yields only a 46.4% F1 score for organization-level SIC code classification, and just 2.1% for healthcare provider taxonomy classification. These results suggest that GPT-4o mini has not memorized SIC or healthcare taxonomy codes from its pre-training data, indicating that this is not a trivial lookup task even for a large language model.

Adding context to the prompt yields mixed outcomes, as shown in Table 2. For SIC code classification, *GPTSum* improves F1 to 49.7%, while *LLaMASum* offers no gain. Similar trends appear in the healthcare taxonomy task, where context slightly helps but overall accuracy remains low. Using *GSnip* provides the largest boost, reaching 60.1% and 25.6% F1 for SIC code and healthcare taxonomy code classification, respectively. These results show that prompting alone is insufficient; however, adding context improves performance by complementing the model's pre-trained knowledge. Nevertheless, the overall performance remains poor, indicating the need for new and more effective methods

5.2. Results of Our Framework

Table 3 presents the performance of four language models fine-tuned using different types of text. Across all architectures, BERT, RoBERTa, Long-

Model$_{Context}$	SIC Code			Healthcare Taxonomy		
	P	R	F1	P	R	F1
BERT						
LLaMASum	0.490	0.513	0.480	0.106	0.185	0.121
GPTSum	0.562	0.560	0.532	0.143	0.173	0.118
GSnip ★	0.700	0.699	0.699	0.555	0.508	0.485
GSnip + LLaMASum	0.723	0.728	0.699	0.723	0.662	0.672
GSnip + GPTSum	0.724	0.718	0.700	0.738	0.679	0.687
RoBERTa						
LLaMASum	0.525	0.512	0.483	0.096	0.175	0.092
GPTSum	0.581	0.565	0.550	0.222	0.189	0.139
GSnip	0.757	0.748	0.741	0.688	0.608	0.604
GSnip + LLaMASum	0.772	0.755	0.750	0.718	0.690	0.697
GSnip + GPTSum	0.774	0.769	0.763	0.742	0.683	0.693
Longformer						
LLaMASum	0.512	0.515	0.481	0.055	0.069	0.049
GPTSum	0.586	0.586	0.581	0.162	0.169	0.089
GSnip	0.754	0.754	0.746	0.707	0.634	0.643
GSnip + LLaMASum	0.740	0.727	0.716	0.714	0.685	0.691
GSnip + GPTSum	0.767	0.758	0.750	0.693	0.689	0.694
GPT-4o-mini						
LLaMASum	0.657	0.648	0.649	0.187	0.219	0.191
GPTSum	0.673	0.663	0.664	0.249	0.237	0.221
GSnip	0.823	0.818	0.817	0.742	0.725	0.728
GSnip + LLaMASum	**0.827**	**0.825**	**0.823**	**0.745**	**0.726**	**0.729**
GSnip + GPTSum	0.826	0.822	0.822	0.717	0.703	0.705

★ Results for BERT trained with *GSnip* for the SIC code classification task are taken from (Jiang et al., 2023).

Table 3: Performance of fine-tuned models using different types of text on two classification tasks from distinct domains. (i) The *SIC Code* section reports results for organization-level SIC code classification, and (ii) the *Healthcare Taxonomy* section reports results for healthcare provider taxonomy code classification. Performance is measured using macro-averaged precision, recall, and F1-score. The best-performing model, GPT-4o mini, trained with *GSnip+LLaMASum*, is shown in bold.

former, and GPT-4o mini, the models trained with either (*GSnip + GPTSum*) or (*GSnip + LLaMASum*) outperform those trained with individual text sources (*GSnip*, *GPTSum*, or *LLaMASum*). Our overall best-performing model is GPT-4o mini, trained with *GSnip+LLaMASum*, bolded in Table 3, achieving 82.3% and 72.9% macro-averaged F1 scores for the SIC code classification and healthcare taxonomy classification tasks, respectively. Even the simple BERT model trained with *GSnip+GPTSum* and *GSnip+LLaMASum* outperforms BERT trained only with *GSnip* by 20.2 and 18.7 points in the healthcare taxonomy classification task. For the SIC code classification task, the results are comparable. This trend confirms that combining retrieved and generated text provides complementary information that enhances classification performance regardless of the model type. Among the encoder-based models, RoBERTa with *GSnip+GPTSum* achieves a 76.3% F1 score for the SIC code classification task, while Longformer achieves a 69.4% F1 score for the healthcare taxonomy classification task. These represent the best-performing encoder-based models following GPT-4o mini.

When comparing Table 2 and Table 3, it is evident that fine-tuning with text acquired by our framework substantially outperforms all prompting-based approaches, demonstrating the overall efficiency of our framework. Our best-performing model (bolded in Table 3) further outperforms the best prompting result, GPT-4o mini with *GSnip*, by 22.2 points for the SIC code classification task and 47.3 points for the healthcare taxonomy classification task.

With *GSnip+LLaMASum*, GPT-4o mini achieves the best F1 scores across both tasks, while *GSnip+GPTSum* shows a slight drop of 2.3 points in the healthcare taxonomy task compared to *GSnip*. This suggests that when fine-tuned on its self-generated summaries, the model may become biased toward its prior knowledge, but by leveraging the strengths of both web-retrieved and LLM-generated text, the overall performance remains strong. Furthermore, the results remain robust irrespective of which LLM is used for summary generation. Both *GPTSum* and *LLaMASum* contribute valuable information when combined with *GSnip*, yielding improved performance across different models in both tasks.

These findings show that our framework is model-agnostic and summary-source-independent: it reliably improves classification performance whether we use GPT or LLaMA-based summaries and whether the underlying classifier is an encoder or a generative model. Overall, the consistent improvements across architectures and domains validate the effectiveness and generalizability of our proposed framework.

6. Analysis

We analyze the SIC code classification task as a representative case study to gain insights into the performance and design choices of our framework.

6.1. Why do Google Snippets outperform LLM summaries?

Our first analysis investigates why Google snippets performed better than LLM-generated summaries (specifically, *GPTSum*). We manually looked at 25 instances for which *GPTSum* led to an incorrect label but *GSnip* produced the correct label.

In four cases, GPT-4o mini did not produce any summary (e.g., *"I don't have current detailed information . . . "*), leaving the model with no usable context so predictions were essentially arbitrary. However, Google returned information that supported the correct label. This finding shows that LLMs cannot always provide information about specific real-world entities if those entities are rare.

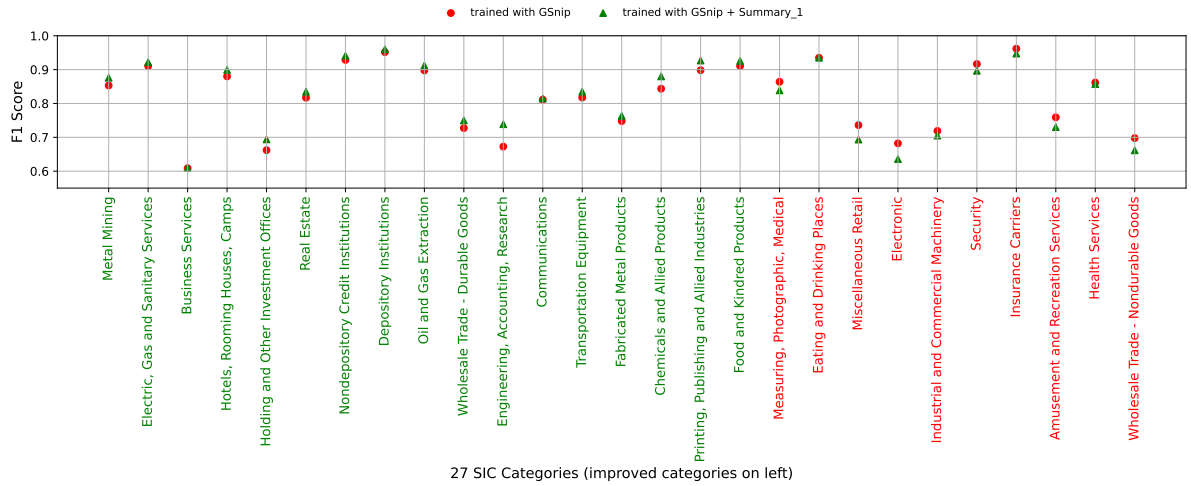


Figure 5: F1 scores across 27 SIC categories for GPT-4o mini fine-tuned with *GSnip* vs. *GSnip+GPTSum*.

The most interesting issue pertains to the semantic framing of LLM-generated summaries. The summaries sometimes mentioned many things and included highly relevant information alongside tangential information. For example, *GPTSum* emphasized secondary activities for *SPARTON CORP*, such as medical instrumentation, resulting in a *Measuring, Photographic, Medical (9)* label instead of the correct *Electronic (7)* label. Similarly, *AmREIT Monthly Income & Growth Fund IV LP* was misclassified as *Real Estate (20)* due to the absence of financial and investment context, which was present in *GSnip* and led to the correct label *Holding and Investment Offices (21)*.

Overall, the summaries produced by GPT-4o mini were coherent and fluent but often presented a broad narrative that omitted the most relevant operational information. In contrast, *GSnip* returned Web content that usually focused on the most salient facts. Aggregating 10 snippets also provided more semantically diverse content that enhanced robustness.

6.2. Performance across Categories

Figure 5 shows the F1 scores for each SIC category individually, for two of our models: GPT-4o mini using *GSnip* or using *GSnip+GPTSum*. The X-axis displays the categories, and the Y-axis shows the F1 scores. Each category is color-coded: green indicates categories where the combined text (*GSnip+GPTSum*) outperforms *GSnip* alone, and red indicates where *GSnip* yields better performance. These results show that 17 of the 27 categories benefitted from using both the LLM-generated summaries and the Google snippets.

Overall, performance varies across categories. Some lower-performing categories are challenging due to their generality (e.g., *Miscellaneous Retail* and *Business Services*), while others require fine-grained semantic distinctions (e.g., there are distinct categories for Wholesale Trade of Durable vs. Nondurable Goods).

6.3. Why Top-10 Google Snippets?

We also evaluated performance using different numbers of Google snippets to assess their impact on the results. Table 4 shows that using multiple snippets improves performance up to a point: the F1 score increases steadily from top-1 to top-10, peaking at **0.817**. Beyond that, additional snippets provide diminishing returns and may even degrade performance, likely due to noise or redundancy.

Fine-tuned GPT-4o mini	P	R	F1
Top-1 snippet	0.691	0.687	0.686
Top-5 snippets	0.771	0.768	0.768
Top-10 snippets	**0.823**	**0.818**	**0.817**
Top-15 snippets	0.779	0.776	0.777
Top-20 snippets	0.813	0.811	0.810

Table 4: Performance of GPT-4o mini across different numbers of retrieved Google snippets.

6.4. High Precision Categorization

Ultimately, our goal is develop classification models that can be used to rapidly populate knowledge resources with task-specific information about real-world entities. For SIC code classification task, our best classification model achieved 82.7% precision, which is good. But ideally we would like even higher precision so that some instances can be labeled fully automatically without the need for human inspection.

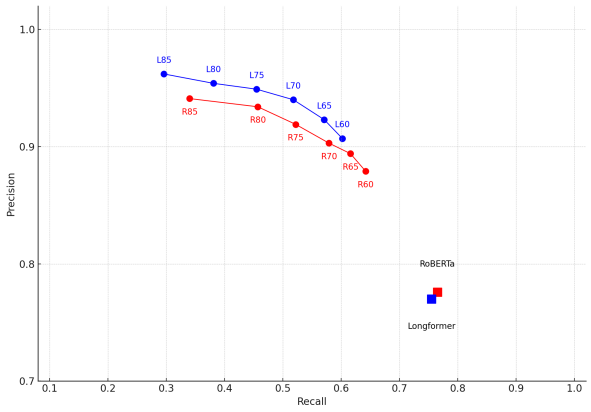


Figure 6: Precision–recall trade-off curves for Longformer and RoBERTa using confidence thresholds.

Toward this end, we experimented with confidence thresholds to effect a trade-off between precision and recall. This approach considers the confidence of each label predicted by the classifier and only assigns the label if the confidence exceeds a threshold. Our best classifiers used GPT-4o mini, but it does not provide built-in support to extract confidence scores or log probabilities. So we ran these experiments with RoBERTa and Longformer.

Figure 6 displays the results. The squares in the lower right corner show the performance of RoBERTa and Longformer when labeling every instance (no threshold). The other data points show their performance using threshold values ranging from .60 to .85 in increments of five. Longformer's confidence scores proved more reliable than RoBERTa's, producing stronger high-precision results.

This strategy successfully produced high precision ($>$ 90%) classification while sacrificing some recall. For example, with a threshold of 0.85, Longformer achieved 96% precision with around 30% recall; using 0.60, it reached 91% precision with 60% recall. These high-precision classifiers may support fully automatic knowledge population for the most confident predictions.

7. Conclusion

We introduced a framework that, given only entity names and their corresponding gold labels as input, can automatically generate descriptive text for those entities, which can then be used to train a classifier. The gold labels are provided only for model training and are not used during text acquisition, allowing the framework to operate independently of any pre-existing structured text-based resources. Our approach leverages web search and large language models to automatically acquire task-relevant text. We evaluated our framework on two classification tasks from distinct domains. The healthcare taxonomy code classification task is entirely different from the SIC code classification task, demonstrating the robustness of our proposed framework. Overall, our framework offers a scalable solution for developing entity classification models across diverse real-world tasks, including those involving lesser-known entities.

8. Ethics Statement

All healthcare providers included in our benchmark are based in the United States. We obtained the provider names and their corresponding taxonomy codes from the National Plan and Provider Enumeration System (NPPES), maintained by the Centers for Medicare & Medicaid Services (CMS). The NPPES registry is publicly accessible and downloadable in the U.S., and the data are released by CMS as publicly available records. Healthcare providers submit this information themselves as part of the National Provider Identifier (NPI) registration process, which is required to become HIPAA-covered healthcare providers in the United States.

Our healthcare provider dataset includes Google search snippets retrieved using SerpAPI and large language model (LLM)-generated text. To avoid redistributing third-party content, we do not release the raw Google snippets or the LLM-generated text. Instead, we provide detailed instructions in our GitHub repository describing how the dataset can be reconstructed using the same retrieval procedure, model version, and prompting setup. This approach ensures transparency and reproducibility without redistributing third-party or model-generated content.

All experiments were conducted for research purposes only. The dataset does not contain sensitive attributes beyond publicly available professional information. We believe these measures ensure compliance with scientific integrity standards while minimizing potential ethical risks related to data redistribution.

9. Acknowledgements

This research was supported in part by the ICICLE project through NSF award OAC-2112606. We thank Tianyu Jiang for helpful clarifications on their publicly released codebase, which facilitated our reproduction of the results reported in their work.

10. Bibliographical References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives.

2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, page 1148–1158, USA. Association for Computational Linguistics.

Tianyu Jiang, Sonia Vinogradova, Nathan Stringham, E. Louise Earl, Allan D. Hollander, Patrick R. Huber, Ellen Riloff, R. Sandra Schillo, Giorgio A. Ubbiali, and Matthew Lange. 2023. Classifying organizations for food system ontologies using natural language processing.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *CoRR*, abs/2107.07566.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2023. A survey on deep learning for named entity recognition : Extended abstract. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3817–3818.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. Linden: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 449–458, New York, NY, USA. Association for Computing Machinery.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA. Association for Computing Machinery.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. *Theory and Applications of Categories*.

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

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal. Association for Computational Linguistics.

11. Language Resource References

Jiang, Tianyu and Vinogradova, Sonia and Stringham, Nathan and Earl, E. Louise and Hollander, Allan D. and Huber, Patrick R. and Riloff, Ellen and Schillo, R. Sandra and Ubbiali, Giorgio A. and Lange, Matthew. 2023. *Classifying Organizations for Food System Ontologies using Natural Language Processing*.

MetaAI. 2024. *Meta Llama 3.1 8B Instruct, An instruction-tuned large language model (8 billion parameters) released by Meta AI, designed for multilingual text generation and instruction following.*

NPPES. *National Plan and Provider Enumeration System (NPPES/NPI)*.

OpenAI. 2024. *GPT-4o-mini (2024-07-18), A lightweight variant of OpenAI's GPT-4o model, released on July 18, 2024*.

A. Appendix

```
{
  "messages": [
   {"role": "system", "content": "You are a classifier that assigns SIC codes based on an
organization's name and business description."},
   {"role": "user", "content": "Organization: <org>\nDescription: <business_description>"},
   {"role": "assistant", "content": "<sic_code>"}
  ]
}
```

Figure 7: Example of the train and dev instance format used to fine-tune GPT-4O mini. Each instance consists of a system instruction, a user message combining the organization name and its business description, and an assistant response containing the target SIC code.

```
{
  "messages": [
   {"role": "system", "content": "You are a classifier that assigns SIC codes based on an
organization's name and business description."},
   {"role": "user", "content": "Organization: <org>\nDescription: <business_description>"},
  ]
}
```

Figure 8: Input format used during inference with the fine-tuned GPT-4O MINI model. The model receives a system instruction and a user message containing the organization name and its business description. No gold label is provided during inference.