

Article

AraCoNER: Arabic Complex NER with Gold and Silver Labels

Wesam Alruwaili *, Najwa Altwaijry and Isra Al-Turaiki

College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia;
ntwaijry@ksu.edu.sa (N.A.); ialturaiki@ksu.edu.sa (I.A.-T.)

* Correspondence: wmalruwaili@ju.edu.sa

Abstract

Named entity recognition (NER) is a fundamental task in natural language processing. Recently, non-traditional nouns (known as complex NER) have increasingly emerged, including long noun phrases and ambiguous names, for example, *Birds of Prey (and the Fantabulous Emancipation of One Harley Quinn)*, *Among Us*, and *Chicago*, which may refer to a city or a novel. Such rapidly growing entity names pose significant challenges for NER. Arabic NER research is usually limited to flat and nested entities, overlooking complex entities due to limited resources, the language's rich morphology, and context ambiguity. Such tasks require high-quality annotated data, yet most existing approaches rely heavily on supervised learning, which depends on large amounts of labeled data. However, acquiring large annotated datasets is costly and labor-intensive. We construct our corpus by leveraging the superior performance of large language models (LLMs), which have driven recent advances in dataset generation. We propose an Arabic complex NER (AraCoNER) dataset with semantically ambiguous and complex named entities, using both gold and silver labels. We investigate several agent-based annotation frameworks in addition to the plain LLM to determine the most efficient annotator for our task. Then, we introduce LLMAAA+, an LLM-agent-based framework that integrates an LLM-powered agent as an annotator into an active learning loop to efficiently select what should be labeled. Instead of solely synthesizing the training data from LLMs, we enhance both the annotation and training phases to generate pseudo-labels using k-NN sampling for in-context examples. Such an approach ensures both efficiency and quality, with cost-effective and minimal human involvement. Our results show that combining an LLM (GPT-4) with a structured agent framework (Google ADK) yields the highest annotation accuracy, even with a limited number of annotated examples, supporting the proposed LLM-agent-based active learning framework.

Keywords: active learning; large language model (LLM); LLM-agent-based framework; LLM-powered agents; named entity recognition (NER); natural language processing (NLP)

Academic Editors: Konstantinos Roumeliotis and Nikolaos Tselikas

Received: 9 January 2026

Revised: 7 February 2026

Accepted: 9 February 2026

Published: 10 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Named entity recognition (NER) is a natural language processing task that identifies and extracts specific types of entities from text, such as the names of people, organizations, and geographic locations, as well as expressions of time, currency, and percentages. NER is a widely studied topic in natural language processing and text mining due to its crucial role as a pre-processing step across a range of applications, including information

retrieval [1], question answering [2], and text summarization [3]. Research has indicated that approximately 71% of search queries contain at least one named entity [4]. Recognizing named entities in search queries would therefore provide better understanding of user intent, thus improving search results and enhancing user experience in downstream applications (e.g., query recommendation) [5].

Named entities often have a complex structure, in which an entity is semantically ambiguous and can belong to multiple entity types, making them harder to recognize. In Arabic, NER tasks have provided insights into the versatility of flat (simple) and nested (hierarchical) NER tasks. For instance, in flat NER, “The University of California” could be represented as a single Organization entity, but in nested NER, it would still be an Organization, but it could also contain a nested entity “California” labeled as a Location. Limited research has focused on complex named entities, which include non-traditional nouns such as creative works, full clauses, infinitives, and gerunds/Masdar. Ashwini et al. [6] describe entities such as movies, books, songs, and TV shows as non-traditional nouns that are harder to recognize. Malmasi and Meng [7,8] illustrate complex named entities related to creative works (songs, movies, software, and book names), products (Apple), long noun phrases (The Englishman Who Went Up a Hill But Came Down a Mountain), infinitives (To Catch a Thief), full clauses (“Dear future husband”), and gerunds (Singin’ in the Rain). Equivalents in Arabic include the TV show name *من كل فج عظيم*, full clause *صباح الخير ياعرب*, long noun phrase *الله صلى الله عليه وسلم*, *رسول*, and gerund/Masdar *العلم والايمل*. Corpora enriched with complex names (e.g., creative works, product names, short contexts, and long entity names) are needed to assess model performance in such cases.

As noted in [9] and [10], present-day NER faces several difficulties that may not be present in Arabic NER, such as (1) the context of the entity, (2) the structural complexity of the entity, (3) robustness to shifts in domain, and (4) annotation cost and inconsistency.

1. **Contextuality:** Most Arabic NER benchmarks have focused on domains such as Wikipedia, which typically feature longer sentences and entities containing multiple entity references [11,12]. Some applications receive brief inputs that offer limited context. Datasets featuring such long entity boundaries and low-context settings are needed to evaluate model performance.
2. **Complexity:** Arabic NER data have been collected only from encyclopedias or news domains, which tend to be characterized by a simple syntactic structure (e.g., proper nouns) and a narrow range of entity types representing people, organizations, and locations, although some entities are non-conventional and involve more complex linguistic structures.
3. **Adaptability:** A robust Arabic NER model should be capable of performing well across several domains, including well-structured text, long and short sentences, emerging data, unseen entities, and a variety of registers: classic, Modern Standard Arabic (MSA), and colloquial. Most NER benchmarks, however, focus solely on data from a single domain, limiting the study of domain adaptation. Moreover, current NER models rely too heavily on memorized training entities instead of generalization, leading to poor recognition of unseen entities.
4. **Annotation Bottleneck:** Most NER models are built using supervised machine learning methods and are heavily reliant on the quantity and quality of labeled data. Data annotation, however, remains a costly and labor-intensive process. Several researchers have demonstrated inconsistent annotations in benchmarked complex NER datasets (i.e., MultiCoNER v2 [13]) [14,15]. One major issue occurs when the same named entity recurs more than once in a sentence, but its annotations differ. For example, in the sentence “rice-long, medium, or short-grain white; also, popcorn rice,” the first occurrence of “rice” is annotated as Product, while the second occurrence of “rice” is labeled as Other. These types of incorrect annotations cause ambiguities in

supervised approaches. An alternative approach that reduces the amount of human intervention is needed.

We present an Arabic complex NER (AraCoNER) corpus (Data are available at <https://github.com/AraCoNERI/Dataset>) (accessed on 7 January 2026) to address the difficulties outlined above. Recently, large language models (LLMs) have offered promising ways to automate aspects of information retrieval [16]. They exhibit exceptional knowledge across various domains due to their enormous pre-training datasets [17]. Hence, we adopted active learning, which seeks to reduce annotation efforts by strategically selecting which examples to annotate and ensuring the quality and precision of the process. We explored a wide range of contexts to improve the understanding of entities in various contexts.

Our corpus was designed to address the challenges shown in Table 1 to produce gold labels (we refer to human- and LLM-labeled samples as gold and silver data, respectively). We developed the LLMAAA+ framework, which uses an LLM-powered agent as an annotator to produce silver labels. Specifically, we initialized task-specific models by training them on seed-labeled data, providing a foundation for subsequent fine-tuning, investigated several agent-based annotation frameworks, adopted the best-performing one to produce pseudo-labels via prompt engineering guided by exemplar selection through k-nearest neighbor (k-NN) for K-shot learning, and then applied an active acquisition mechanism to ensure more informative data selection. This approach combines both efficiency and quality, offering a cost-effective yet high-performing solution.

Table 1. Difficulties addressed by the AraCoNER dataset.

Difficulty	Description
Complex	Some named entities can have complex linguistic forms and are difficult to identify, e.g., انتشأ □ سلسلة ملحمة الشفق من الغرق البية البواب وانقلاب الهرم الاجتماعي دائما ستكون □ هناك مقارنه بين محال والنبي وعز ازيل
	Some named entities are context-dependent and may be recognized as named entities in some contexts but not in others, e.g., تدور أحداث اللهم إني صايم حو □ علي الذي يتعرض للظلم ويتم فصله من عمله فيتجه لكسب المال بشتي الطرق ومنها النصب والاحتيا □ أرز باللين لكل من اراد التفاو □ والاستمتاع بحياته كتاب جميل وإسلوب رحاب بسام أكثر من رائع أغلب مقالاته كانت مميزة جدا بإسلوبها ومضمونها أنا عشقت، جميلة جدا جميلة بشكل مؤلم من فرط إندماجي حلمت بكوابيس متعلقة بأبطل □ الرواية السادس من أكتوبر وتعتبر من أهم المناطق الراقية في محافظة الجيزة
Emerging	Some groups of entity types experience rapid growth, which may be released weekly, e.g., new books, songs, and movies. Such entities can also pose difficulties for human annotators, as they may not be familiar with the most recent media releases.

The central objective of this study is to generate high-quality silver annotations using LLM-powered agent annotators, given access to a strong gold seed. We summarize our contributions as follows:

- We introduce the AraCoNER corpus to address the difficulties discussed above. Our corpus was collected from multiple domains (Wikipedia, news, blogs, and online reviews) that deal with MSA and spoken Arabic, including complex names in short contexts, long names, and emerging entities.
- We demonstrate an efficient and cost-effective annotation strategy for building AraCoNER by leveraging LLM-powered agents within the LLMAAA+ framework, requiring minimal human intervention while maintaining high annotation quality.

The remainder of this paper is organized as follows: We first review related work, and then introduce the ARACONER dataset and outline the proposed methodology. Next, we present the experimental results, followed by an analysis and report of the overall findings, and finally, we conclude the paper.

2. Related Work

2.1. Arabic Named Entity Recognition (NER) Corpora

Most Arabic NER datasets are designed to annotate flat NER. For instance, ANER-Corp [18], collected from news sources, includes a dataset of 150 k tokens divided into four entity categories. Another corpus, AQMAR [12], was collected from Wikipedia and covers domains such as history, science, technology, and sports. It is annotated with open-ended entity types. In addition, a classical Arabic corpus, CANER-Corpus [11], was derived from the Islamic Hadith context and includes approximately 258 k tokens annotated for 14 entity types. Moreover, Ontonotes5 [19] contains approximately 300 k tokens sourced from newswire and conversational telephone speech across Arabic, Chinese, and English.

Other recent Arabic benchmark datasets support both flat and nested entity annotations. The largest Arabic NER corpus, Wojood2022 [20], was introduced to support MSA and covers 550 k tokens annotated for 21 entity types. An extension of this dataset is called Wojood2024 [21], which addresses the limitations of the previous coarse-grained entity types, such as PERS, ORG, and LOC, by providing entity subtypes, totaling 47 k tokens covering 51 subtype entity mentions.

2.2. Complex NER Corpora

The MultiCoNER series of datasets primarily focuses on identifying complex named entity types. MultiCoNERV1 [9] contains 13 subsets: 11 monolingual, 1 multilingual (MULTI), and 1 code-mixed (MIX). The multilingual track is a subset of the data in which samples from all languages are randomly interleaved, and the code-mixed track includes instances in which the entity belongs to a specific language while the surrounding text is written in a different language. MultiCoNERV1 is divided into six categories: PEOPLE, LOCATION, CORPORATIONS, GROUPS, PRODUCTS, and CREATIVE WORKS. MultiCoNER V2 [13] covers 12 languages and a multilingual subset. It is classified into 33 fine-grained classes, grouped across six coarse types: PEOPLE, LOCATION, GROUPS, PRODUCTS, CREATIVE WORKS, and MEDICAL. In total, 30% of the test sets in the monolingual track are corrupted with simulated noise to improve robustness.

2.3. Large Language Models (LLMs) for NER

NER has drawn considerable interest from researchers seeking to explore new opportunities in the era of LLMs [10,22–25]. To bridge the gap between sequence labeling and generation, Wang et al. [26] proposed GPT-NER to transform NER into a generative task and introduced a self-verification mechanism to correct NULL inputs as entities. Instead of explicitly identifying entities, the task of recognizing a location entity such as “Chicago” is reformulated as generating a revised sentence: “@@Chicago## is a city,” where the special tokens “@@” and “##” indicate entity boundaries. This enables LLMs to handle sequence labeling in a more natural text generation format, yielding promising results, particularly in scenarios with few training examples and low resources. Similarly, Xie et al. [27] introduced a training-free, self-improving framework that uses LLMs for zero-shot NER, improving performance by generating pseudo-demonstrations from unlabeled corpora.

Another notable development in NER involves the use of few-shot learning techniques, which enable models to learn effectively from only a small number of samples. Ashok et al. [28] introduced PromptNER to demonstrate how entity type definitions in prompts enable LLMs to identify entities and provide supporting explanations. In addition, Hu et al. [29] demonstrated ChatGPT's capability in few-shot clinical NER scenarios through iterative refinement and prompt assessment. Such step-by-step refinement helps identify and address any flaws or limitations in the prompts, ensuring they remain effective and relevant over time. In a further advancement of few-shot learning, Zhu et al. [30] proposed GL-NER as a generation-aware LLM for few-shot NER, allowing it to generate entity labels and indicate "does not exist" when no entity is found. Moreover, they tackle the underlying difficulties of few-shot NER, where labeled examples may be limited by applying a masking-based loss optimization strategy.

Another study explored advanced data augmentation techniques to boost the effectiveness of complex NER in low-resource languages such as Bangla, Hindi, Spanish, and Turkish. Zhang et al. [31] proposed E2DA to integrate both exogenous and endogenous data augmentation. The former approach uses LLMs to generate additional data guided by specific prompts, thereby enhancing the dataset's diversity. The latter approach exploits semantic relationships within the existing data to better exploit meaningful features.

2.4. LLM-Based Annotation

Researchers have investigated the use of LLMs for data labeling across a variety of natural language processing tasks such as hate speech detection [32], political tweet classification [33], sentiment analysis [34], topic classification [35], named entity recognition [36], and question generation [35], using datasets in English, French, Spanish, Italian, and Basque. Several studies have shown that LLMs can outperform human annotators and crowd workers as data annotators [37,38]. Törnberg et al. [33] showed that GPT-4, when used in a zero-shot setting, surpasses human annotators in labeling English language political tweets. Further analysis revealed that GPT-4 can accurately label tweets that require logical reasoning based on contextual information. Li et al. [39] explored the use of ChatGPT to annotate hateful, offensive, and toxic comments and compared the results with MTurk annotations. They found that ChatGPT's performance is promising, as it correctly labels 80% of the comments. Moreover, its performance was more consistent in labeling non-harmful comments compared to harmful ones.

Techniques such as data augmentation, transfer learning, and active learning can enhance the quantity and quality of training data for LLMs, even with incomplete or noisy data and in low-resource languages [40]. Such techniques can rapidly label large-scale datasets, reducing the time of tasks that typically rely on high-cost human annotation. Chen et al. [41] introduced a novel approach to use LLMs as expert annotators. They assessed the direct use of LLMs for event detection and event argument extraction, identifying a significant performance gap compared to fine-tuned models. To address this gap, they employed LLMs as expert annotators to produce labeled data consistent with existing datasets and subsequently fine-tuned specialized event extraction models with the augmented data. Such an approach enhances the performance of fine-tuned models by reducing data scarcity and imbalance.

LLM-based annotation can be affected by input variations, as even minor changes in prompt formulation or model sampling parameters may result in inconsistent annotations [42,43]. Ensuring consistency in LLM-generated annotations remains difficult, necessitating further research into prompt engineering and hybrid validation strategies that involve both humans and LLMs.

2.5. LLM-Based Active Learning

Active learning is a widely studied approach that seeks to minimize data labeling effort by selecting the most informative data samples for annotation and subsequent model training [44,45]. A new paradigm has emerged with the advent of LLMs, which exhibit notable reasoning and generation capabilities [46,47]. Instead of relying exclusively on predefined metrics such as uncertainty or diversity to select data instances, LLMs can now identify the most informative samples through reasoning or even generate entirely new instances, thereby enhancing the efficiency of model training [48,49]. By collecting these informative samples, LLMs can introduce new data annotation approaches by either collaborating with a human labeler or simulating them directly, thereby minimizing manual annotation effort [50]. Furthermore, Wang et al. [35] proposed an approach based on active learning that leverages both human and GPT-3 labels, leading to better fine-tuning performance.

2.6. Human–LLM Collaboration

Although LLM-based annotation has greatly lowered costs, it remains prone to errors, especially in complex or domain-specific tasks [51]. To tackle this issue, researchers have developed methods to assess annotation quality and dynamically direct data to either human or LLM annotators, balancing both efficiency and reliability. For instance, Wang et al. [50] introduced a multi-step human–LLM collaborative approach in which LLMs initially generate labels and explanations. The labels are then evaluated by a verifier and human experts to re-annotate a portion of low-quality cases. Another human–LLM collaboration approach lets humans curate annotated examples for use in LLM prompts, enabling annotation in a few-shot learning scenario. Although such an approach is straightforward and efficient, selecting optimal examples for few-shot learning remains difficult.

To bridge the gap between humans and LLMs, Fernandes et al. [52] illustrated guard-rail tuning: training models to avoid controversial statements, which can pose difficulties for certain annotation tasks, as the models may refuse to annotate sensitive topics, such as the association between COVID-19 vaccine resistance and individual U.S. political parties [33]. Custom fine-tuning—training the model with labeled data designed for a specific annotation task—can further enhance text annotation performance and achieve excellent results, even with only 50–100 manually labeled samples.

Despite the progress highlighted in the reviewed literature, existing Arabic NER datasets primarily focus on flat and nested NER, or on specific dialects or domains where named entities are simple names. Thus, their use leads to language models that are sensitive to domain-specific variations. The current complex NER datasets are for languages other than Arabic. Table 2 presents the benchmark datasets for NER and compares them with ours (in bold). Most current NER models rely primarily on supervised learning and are highly dependent on the size and quality of annotated datasets, while annotating data remains labor-intensive and costly. Notably, recent approaches to Arabic NER and complex NER often involve fine-tuning a language model. A long-tailed dataset in other languages can be fine-tuned well with commonly occurring entities, but some difficulty is encountered when classifying rare entities. Current research has demonstrated that LLMs offer promising avenues for advancing NER across various domains, aiming to leverage the strengths of both humans and LLMs for optimal annotations while minimizing cost and human effort.

Table 2. Named entity recognition benchmark datasets, where MSA refers to modern standard Arabic, classic refers to the historical, literary, and religious forms of Arabic, and dialect refers to spoken varieties of Arabic.

Corpus	Complex	Arabic	Label	Tokens	Entities	Category	Domain
ANERCorp	No	MSA	Gold	150 k	11 k	4	Website, news, and magazines
AQMAR	No	MSA	Gold	74 k	–	Open	Wikipedia
Ontonotes5	No	MSA	Gold	300 k	28 k	18	News
CANER-corpus	No	Classic	Gold	258 k	72 k	14	Religion
Wojood	No	MSA-Dialect	Gold	550 k	75 k	21	Web articles, archive, and social media
Wojood-fine	No	MSA-Dialect	Gold	550 k	47 k	51	Web articles, archive, social media and news
MultiCoNERV1	Yes	No	Gold	–	–	6	Wikipedia, questions, search queries
MultiCoNERV2	Yes	No	Gold	–	–	33	Wikipedia
AraCoNER	Yes	MSA+ Dialect	Gold + Silver	>213 k	>24 k	7	Wikipedia, news, blog, reviews

3. AraCoNER Dataset Overview

The AraCoNER dataset was developed to tackle the difficulties in NER outlined in Section 1. It establishes a comprehensive corpus exhibiting ambiguous semantics and a high level of complexity. It targets a variety of Arabic websites and provides up-to-date information, including Wikipedia, news, reviews, and blog articles in Arabic. The dataset is rich in complex names (e.g., emerging entities, creative works, product names, short contexts, and long entity names) to improve data complexity. Malmasi et al. [9] applied sentence length filtering to retain short sentences, where samples shorter than 28 characters and longer than 180 characters were excluded. However, such a method may omit critical named entities. For example, “Apple launched iPhone 15” is a short and informative sentence of 25 characters that includes two named entity types. Identifying a minimum sentence length is difficult, so we rely on collecting data from a variety of sources covering different sentence lengths. Lastly, we employed two annotation methods: gold-standard annotations (manually labeled by humans) and silver-standard annotations (generated by LLMs).

3.1. NER Taxonomy

AraCoNER adopts the entity type taxonomy from WNUT 2017 [53], which defines an NER tag set consisting of six categories. We extend this tag set by introducing an additional category, Medical, to increase diversity. Therefore, we have seven named entity tags:

- PERSON (PER, names of people);
- LOCATION (LOC, locations/physical facilities);
- GROUPS (GRP, all other groups);
- CORPORATION (CORP, corporations/businesses);
- CREATIVE-WORK (CW, movie/show/song/book titles);
- PRODUCT (PROD, consumer products);
- MEDICAL (MED, disease/medication/vaccine/symptoms).

3.2. Domain and Data Sources

AraCoNER was collected from multiple sources across several domains, covering a wide range of topics. A subset of the data was manually collected from online articles published on Wikipedia, news websites, and blogs covering several topics, including health, technologies, sport, media content, finance, investment, communities, cultures, and consumers in MSA. We believe Wikipedia is a valuable source for historical data (e.g., older movies, songs, products, person names), while news articles provide up-to-date information (e.g., recent releases). Another subset was extracted from sentiment analysis corpora [54,55]. These corpora cover customer reviews across several domains: restaurants, products, tourism, and books. We adopted these datasets to address spoken (dialectal) Arabic text within the scope of our taxonomy. Moreover, such data may contain noisy text (e.g., typing errors due to keyboard layout), allowing us to evaluate the NER model's resilience to noise. We believe that when consumers discuss tourism, they typically refer to locations, sports teams, and music groups; when discussing restaurants, they focus on corporations and food products; and when writing about books, they focus on creative works and authors. Therefore, using tourism and book data, we can identify the target named entity categories: person, location, group, corporation, product, and creative work. Rather than using the entire corpus of these sentiment analysis datasets, we selected a random subset from each dataset for our corpus. We aimed to generate a corpus containing both MSA and dialect texts from various sources.

We obtained our data samples by parsing all articles from Wikipedia, news, blogs, and reviews into sentences. We eliminated all noise, such as extra characters, samples, non-Arabic characters, and URLs, to provide a clean Arabic text without altering its meaning or content. During the cleaning process, we frequently encountered cases—especially when dealing with foreign books, songs, movies, or even diseases—where the entire sentence was written in Arabic but the named entity itself appeared in English, for example, “أعتقد أعراض Covid بدأت تظهر” or “أعظم كتاب قرأته في حياتي” *The Great Gatsby*. In such cases, we chose to remove the entire sentence, as the most significant part—the named entity—was not in Arabic and thus not useful for our task.

The dataset was created by the authors for this study and it includes no copyrighted or access-restricted third-party content. The authors explicitly allow the data's free use and redistribution for research purposes.

3.3. Annotation Guidelines and Process

We constructed a dataset of over 10 k samples comprising more than 213 K tokens. The dataset was derived from two main subsets. The first subset consists of texts manually collected from Wikipedia, news websites, and blogs. The second subset was drawn from existing sentiment analysis corpora, including customer reviews. We assigned around 5600 samples with 98,722 tokens as human-generated labels (gold-labeled data), including all samples from the first subset and a randomly selected, representative portion of the second subset. The remaining samples were annotated using LLMs and treated as LLM-generated labels (silver-labeled data).

For gold data, each sentence was assigned a sentence ID, and its tokens were arranged vertically. Thus, every token was assigned a position index reflecting its order within the sentence. The resulting corpus was saved in comma-separated values (CSV) format with four columns (sentence ID, token position, token, and tag) and uploaded to Google Sheets for annotation. Our annotation guidelines were designed to be compatible with the WNUT2017 [53] guidelines, which are based on various NER annotation schemes such as CoNLL [56], ACE [57], and MSM [58]. However, we made certain modifications, such as adding a new tag (medical), and adapted the WNUT2017 guidelines to be more suitable for Arabic names. For instance, we treat prefixes and suffixes as parts of entity

names, such as الملك فيصل, الملكة إليزابيث الثانية. The following list represents the guidelines with examples for each category.

- Creative Work: Names of creative works. They must be created by a human. Example: طاش ما طاش، نسيم علينا الهوى، موسم الهجرة الى الشمال.
- Product: Names of products, real or fictional. They must be something one can touch. Example: ايفون ١٥، السجادة السحرية.
- Medical: Names of diseases, symptoms, medications, and vaccines. Example: السكري، من النوع الثاني، تساقط الشعر، مونجارو.
- Group: Groups of a specific and unique name. Example: نادي الهلال، فرقة ميامي، جامعة الملك سعود.
- Corporation: Names of corporations/business companies. Example: مينا بلات فورم.
- Person: Names of people, real or fictional. Example: أحمد بن الحسين بن الحسن بن عبد الصمد، الجعفي، شوارزنجر، هاري بوتر.
- Location: Names of locations, real or fictional. Example: نهر النيل، أرض نارينا.

We use the BIO tagging schema since it is widely used in NER tasks. In the schema, (B) indicates the beginning of a named entity, (I) indicates tags within the entity, and (O) indicates tags outside the entity (i.e., non-entity words). Several works have adopted BIO-based annotation schemes for annotating complex entities [59–61]. Such schemas have yielded better performance than other annotation schemes for complex entities.

The annotation process was carried out in two stages. In the first stage, we held a Zoom meeting to introduce the annotation guidelines and train around six prospective annotators. All annotators independently annotated the same set of samples (1000 tokens) that was designed to be difficult yet comprehensible. The three participants who scored above 90% were selected to move to the second stage.

In the second stage, the three annotators were given the annotation task. All annotators were qualified with doctoral degrees in Arabic. Supervised by a former expert annotator, the annotation process lasted three months. First, each annotator was assigned approximately 10 k tokens for annotation. During this phase, the expert helped with doubts and complex cases and provided ongoing feedback to the annotators. For example, the annotators were sometimes confused between entities referring to groups and entities referring to religious communities and events. The expert reviewed all annotations and asked the annotators to revisit certain cases. A program was developed to automatically extract and review all entities in case of errors (e.g., category names) or missing annotations (e.g., Nan). Again, in cases with errors or incompleteness, the expert provided targeted feedback to the annotators. The main goal of this stage was to enhance annotation consistency across annotators. The process was repeated three times with a new set of tokens, and each annotator labeled a total of 31 k tokens. Table 3 shows the number of entities in each category for gold and silver labels.

3.4. Inter-Annotator Agreement

To measure annotation quality, we randomly selected 3000 tokens from each annotator that had been previously annotated by another annotator. This resulted in pairwise inter-annotator agreement (IAA) comparisons, such as A1–A2, A2–A3, and A3–A1, totaling approximately 9000 tokens (9.1% of the entire corpus).

Table 3. Counts of entities per class for gold and silver labels.

Class	Gold Labels	Silver Labels
PER	1944	2593
LOC	1302	3569
GRP	480	2043

CORP	1778	1267
CW	2621	1587
PROD	2131	763
MED	1923	878
Total	12,179	12,700

While IAA is typically measured using Cohen’s Kappa, some researchers argue that it is not well suited for evaluating named entity annotations for several reasons. Kappa statistics and reliability coefficients cannot be calculated without a negative case count, whereas named entities are sequences of tokens with no negative cases [62]. Moreover, due to the class imbalance—where tokens tagged “O” far outnumber those tagged with other labels—Kappa may overestimate agreement scores. However, omitting the “O” label from the calculation can lead to an underestimation of agreement [63,64]. Therefore, we chose not to use the Kappa value, since entity spans were also compared, which poses a difficulty, as the expected chance agreement for each entity type and span can be extremely low [65]. Entities might have different spans (e.g., *ثلاثية غرناطة*, *رواية ثلاثية غرناطة*), so we calculated the IAA through the F-measure. In particular, we calculated the IAA between pairs of annotators by treating one annotator as the gold standard and computing the F-measure, which is the harmonic mean of precision and recall. The F-measure was computed at the token and entity levels (strict-match and relaxed-match F1). In token-level agreement, we measure agreement per token, regardless of entity span boundaries. In entity-level agreement, relaxed match measures agreement on the label and some overlap in span, while strict match measures agreement on both the entity boundaries and the entity label. Tables 4–6 report the IAA scores for each category at the token and entity levels (relaxed and strict), where A1–A2 refers to the F1-score between the first and second annotators.

Table 4. Token-level agreement between each pair of annotators.

Tag	A1–A2				A2–A3				A3–A1			
	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1
PER	85	1	1	0.99	96	0	1	0.1	133	4	5	0.97
LOC	32	3	3	0.91	53	8	7	0.88	45	4	4	0.92
GRP	46	6	1	0.93	42	5	0	0.94	42	8	0	0.91
CORP	71	5	9	0.91	91	8	15	0.89	74	2	0	0.99
CW	147	15	3	0.94	131	6	12	0.94	156	7	11	0.95
PROD	105	4	11	0.93	118	6	15	0.92	100	3	14	0.92
MED	162	11	18	0.92	90	3	9	0.94	92	9	17	0.88
Micro	648	45	46	0.93	621	36	59	0.93	642	37	51	0.94

Table 5. Relaxed F1-score for inter-annotator agreement between each pair of annotators.

Tag	A1–A2				A2–A3				A3–A1			
	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1
PER	40	0	1	0.99	51	0	1	0.99	61	3	1	0.97
LOC	22	1	2	0.94	38	2	1	0.96	30	1	2	0.95
GRP	18	2	1	0.92	20	2	0	0.95	22	3	0	0.94
CORP	47	2	2	0.96	64	4	3	0.95	47	1	0	0.99
CW	72	3	0	0.98	73	2	2	0.97	69	3	2	0.97
PROD	61	2	4	0.95	68	2	7	0.94	56	1	9	0.92
MED	76	4	4	0.95	47	1	2	0.97	48	2	7	0.91
Micro	336	14	14	0.96	361	13	16	0.96	333	14	21	0.95

Table 6. Strict F1-score for inter-annotator agreement between each pair of annotators.

Tag	A1–A2				A2–A3				A3–A1			
	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1
PER	39	1	2	0.96	51	0	1	0.99	59	5	3	0.94
LOC	21	2	3	0.90	35	5	4	0.89	28	3	4	0.89
GRP	18	2	1	0.92	20	2	0	0.95	21	4	1	0.90
CORP	44	5	5	0.90	60	8	7	0.89	47	1	0	0.99
CW	69	6	3	0.94	70	5	5	0.93	66	6	5	0.92
PROD	58	5	7	0.91	65	5	10	0.90	55	2	10	0.90
MED	72	8	8	0.90	44	4	5	0.91	45	5	10	0.86
Micro	321	29	29	0.92	345	29	32	0.92	321	26	33	0.92

4. Methodology

To minimize annotation costs and reduce human intervention, the active learning methodology proposed by Zhang et al. [66] was adopted in this study. These authors optimized both the annotation and training processes by integrating active learning into a vanilla LLM to select more informative data; enhanced annotation quality using an LLM with few-shot prompts while using k-NN examples to generate training data; and improved training with automatic reweighting, assigning learnable weights to the silver training data to emphasize informative samples while mitigating noise. We used the same methodology with some modifications to better suit our research context. We capitalized on the superior few-shot performance of LLMs while leveraging abundant unlabeled data by treating LLM-powered agents as annotators rather than vanilla LLMs to train task-specific inference models.

The main goal of this research is to produce high-quality silver annotations, improving entity boundaries and label consistency, using LLM-powered agent annotators initialized with a gold seed to provide a strong foundation for better generalization. To avoid the risk of self-influence or an echo chamber effect when training a model on silver labels generated by GPT-4, the following steps were taken. We performed a one-time initialization of the task-specific model (TAM) by training it on a substantial amount of gold labels (i.e., seed-labeled data), providing a strong foundation for subsequent fine-tuning. An active acquisition function was then applied to select optimal samples from large unlabeled datasets. Moreover, the LLM-powered agent enhances annotation quality through prompt engineering, using k-NN examples drawn exclusively from human-generated labels rather than LLM-generated labels. Thus, the LLM-powered agent produces pseudo-labels. The automatic reweighting technique was omitted, as its effects were found to be marginal [66]. We repeated the process t times. Figure 1 presents the LLMAAA+ framework, and the prompt design details are provided in Appendix A.

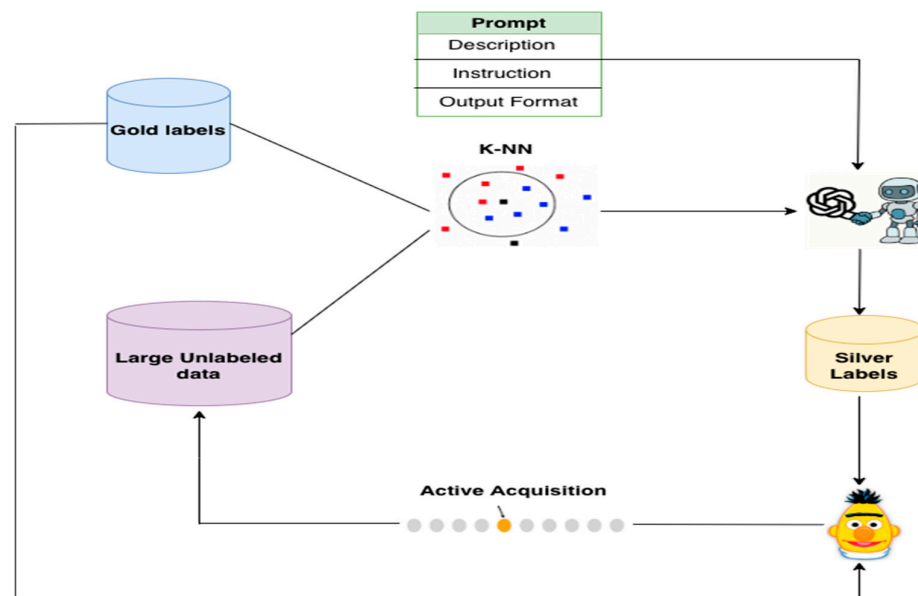


Figure 1. The framework of LLMAAA+.

4.1. Optimizing LLM Agents for Annotation

Although LLMs are powerful, previous research indicates that their effectiveness can vary dramatically depending on the design of task-specific prompts. Kuzman et al. [67] highlighted that LLMs' potential as data annotation tools can be realized with simple prompt design and minimal manual labeling effort. While identifying optimal prompts remains challenging, certain guiding principles have been shown to outperform plain instructions. Following this line of work, we adopted a k-NN retrieval-based dynamic few-shot prompting framework [68]. Specifically, we embedded both the demonstration pool set D_{demo} —consisting of samples drawn from the ground truth data D_{gold} —and the query instance, requiring annotation into dense vector representations. Sentence-level embeddings were generated using Arabic Sentence-BERT (Arabic-SBERT-100K) (<https://huggingface.co/akhooli/Arabic-SBERT-100K>) with mean pooling over token embeddings, and cosine similarity was used to retrieve the top-k nearest neighbors, which serve as in-context exemplars. Semantically similar examples help guide the LLM toward more accurate responses.

These prompting strategies were implemented within an agent-based annotation framework that employs structured prompts, annotating and reflection agents. While the agents share a similar structure, they differ in their internal orchestration, including whether annotation and reflection are executed using a lightweight agent framework (Agno), explicit handoff (OpenAI Agents SDK), or sequential processing (Google Agent Development Kit).

4.2. Selecting Effective Agent Annotators

To mirror our three human annotators' gold labels, we extended the setup by adding three agent-based annotators alongside a plain LLM baseline to generate silver labels. This approach allows us to systematically compare different annotators and determine the most effective one for our Arabic complex NER task.

We first outline the internal workflows, key parameters, and decision-making mechanisms that influence each annotator's annotation behavior. At the most basic level, the plain LLM approach employs a single inference with a fixed prompt and standard decoding parameters. Annotation behavior is entirely prompt-driven, with no built-in

reflection, validation, or correction stages. This motivates the exploration of agents beyond such baseline. The Agno agent relies on a lightweight agent framework characterized by a composable design and low orchestration complexity, building custom logic on top of Agno's unified model interfaces, memory abstractions, and tool integrations. A more structured agentic design is the OpenAI Agents SDK, which utilizes a two-agent architecture with explicit handoffs between an annotating agent and a reflection agent. Schema validation, guardrails, and tracing are applied at each step, with the reflection agent serving as a decision point that can modify or reject annotations prior to finalization. At the orchestration level, the Google ADK uses a sequential two-agent pipeline with annotating and reflection agents, supported by strong typing, built-in safety controls, and validation enforced by the orchestration. This integration limits output variability and ensures corrections before finalization.

We partition the ground truth data D_{gold} into two disjoint subsets, $D_{\text{gold-1}}$ and $D_{\text{gold-2}}$, to avoid data leakage between retrieval and annotation. The rationale is to ensure no overlap between the set used for annotation and the set used for k-NN retrieval. Each of the four annotators (plain LLM, OpenAI, Agno, and Google) independently labeled a set of stratified samples from $D_{\text{gold-1}}$. For each query instance, we retrieved semantically similar examples from $D_{\text{gold-2}}$ using k-NN retrieval to construct few-shot prompts. Based on comparative performance across annotators, the top-performing agent across annotators was selected for the next phase of the study. Figure 2 illustrates the process of selecting the best annotator, and examples of K-NN-based annotation are provided in Appendix B.

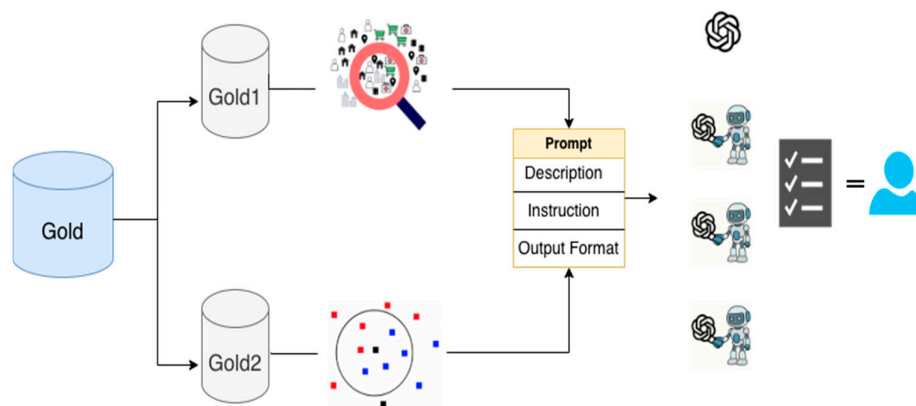


Figure 2. Annotator selection.

4.3. Active Data Acquisition

Active learning aims to reduce annotation costs by strategically selecting examples to annotate. Such strategies typically aim to maximize either uncertainty or diversity.

Uncertainty-based approaches rely on model predictions to select hard examples, whereas diversity-based approaches exploit heterogeneity among selected instances. Following Culotta et al. [69], the least-confidence strategy was used to sort examples according to the probability that model M assigns an item to the predicted class $\hat{y}$, prioritizing those with the lowest confidence:

$$\operatorname{argmax}_{x \in D_{\text{pool}}} (1 - \text{PM}(\hat{y} | x)) \quad (1)$$

Bogdanov et al. [10] found that annotation diversity and the size of the pre-training dataset are key factors influencing LLM performance. We consider a pool-based active learning framework in which a large, unlabeled dataset D_{pool} is available. As a baseline, we employed random selection to sample instances uniformly from D_{pool} .

The active learning process begins with the training of model M on a large seed set D_{gold} . Then, at each iteration, an acquisition function $f(\cdot, M)$ is applied to identify a batch B consisting of b examples from D_{pool} . Least confidence is used to obtain k -NN samples. The LLM-powered agent annotator is queried to label this batch. The labeled batch is then removed from D_{pool} and added to the labeled set D_{silver} . Model M is then trained on b examples of this batch. The process is repeated for t rounds.

5. Experiment and Analysis

5.1. Experiment Setting

Our dataset consists of both labeled and unlabeled data for the Arabic complex NER task. The labeled corpus (gold dataset) contains 5600 samples, of which 1000 were used for training using five-shot learning, 750 for validation, and 850 for testing. The unlabeled dataset used to generate silver labels includes over 7000 samples (172,840 tokens). The sequence length of each instance was between 25 and 512 characters.

5.2. Baselines

We compare LLMAAA+ with the SUPERVISED baseline. We used AraBERT [70] to train the TAM on clean-labeled data D_{demo} used in LLMAAA+.

5.3. Implementation

Several studies have shown how GPT-4 outperforms other LLMs in annotation [66,71]. We therefore employed GPT-4 as the primary LLM and AraBERT as TAM. For the uncertainty-based active acquisition function, we used the average of the maximum token probabilities as the pooling strategy to determine the confidence score for a sequence tagging task.

Phase 1: Annotator Selection. To find the best annotator agents besides plain GPT-4, we randomly split the training set into two equal subsets of 2000 samples for annotation and k -NN retrieval: $D_{\text{gold-1}}$ and $D_{\text{gold-2}}$. Since the dataset in the NER task often contains a long tail of rare entities, we avoided random selection and applied stratified sampling, which selects a subset of the original data frame while preserving the relative distribution of entity categories. We used a stratified sample of 500 instances from $D_{\text{gold-1}}$. Then, each of the four annotators (GPT-4, OpenAI, Agno, and Google agents) independently annotated the 500 instances in parallel. Meanwhile, for each of the 500 query instances, we retrieved semantically similar examples from $D_{\text{gold-2}}$, of which 100 were used, to construct dynamic few-shot prompts via k -NN with $k = 5$. We computed F1-scores at the token and entity levels and reported the micro-F1-score for each annotator relative to human annotations, as detailed in Appendix C. Based on the comparative performance across annotators, Google's Agent Development Kit (ADK) is the best-performing agent.

Phase 2: Active Learning-Based Annotation. Unlike LLMAAA, which reuses the same demonstration and validation sets, we randomly divided the gold set into 1000 samples to form the demonstration set D_{demo} and randomly selected 3000 samples as the seed-labeled set D_{seed} , ensuring that the two sets are disjoint. The validation set D_{val} remained unchanged and comprised 750 instances. For D_{pool} , where silver labeling was required, we relied on the unlabeled corpus.

We initialized the training by fine-tuning AraBERT on a seed-labeled set. At each iteration, model M (fine-tuned AraBERT) predicted on the D_{pool} and selected a batch of 200. Then, the Google agent annotated these 200 samples using k NN few-shot prompting with $k = 5$ from D_{demo} . These 200 new silver-labeled samples were removed from D_{pool} and added to the training set D_{silver} . M model was then retrained on D_{silver} , and the process was repeated 25 times.

For both settings, SUPERVISED and LLMAAA+, TAMs were trained three times using different random seeds, and we report the mean and standard deviation of the outcomes. Implementation details are provided in Appendix D.

5.4. Agent-Based Annotation Performance

To guarantee the performance of Arabic complex NER annotation, we further investigated performance with other annotators, including plain GPT-4 and agent-based annotators implemented using OpenAI, Agno, and Google. Due to budgetary considerations, our experiments were restricted to annotating only 500 samples. The results indicate that the Google agent was the best annotator, based on the average micro-F1- and macro-F1-scores, as shown in Figure 3. The Google agent outperformed the plain LLM by a large margin, with a 2.7% improvement in micro-F1 (0.809 vs. 0.782) and a 4% improvement in macro-F1 (~0.79 vs. ~0.75). The observed improvement is likely influenced by the orchestration capabilities of Google ADK, including multi-step reasoning, reflection-based validation, and error handling, which enhance annotation consistency compared to a single-pass plain LLM.

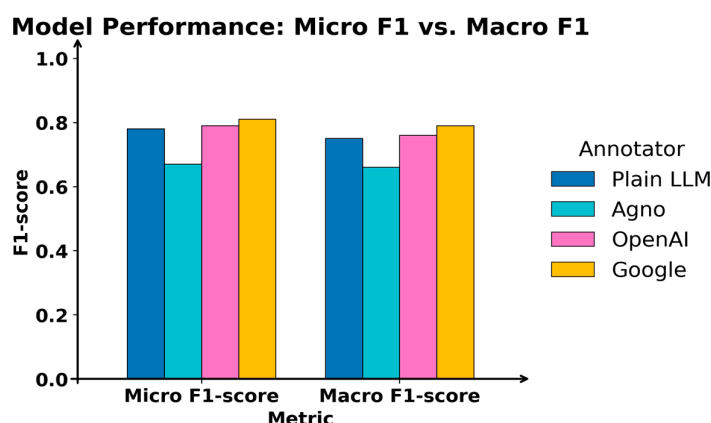


Figure 3. Annotator performance.

Furthermore, the Google agent achieved the highest F1-scores across most entity classes. It performed much better on GRP, obtaining an F1-score of 0.663, while GRP outperformed the plain LLM (0.599), OpenAI agent (0.609), and Agno agent (0.476). Moreover, the Google agent was strong at recognizing and disambiguating LOC, with an F1-score of 0.828, surpassing the plain LLM (0.787) and OpenAI agent (0.817). It was also consistently the best in MED, achieving an F1-score of 0.782, which is higher than that of the plain LLM (0.738), OpenAI agent (0.759), and Agno agent (0.751). For CW, the Google agent had an F1-score of 0.863, which was slightly higher than that of the plain LLM (0.856) and OpenAI agent (0.849) and much higher than that of Agno (0.530). For CORP, the Google agent outperformed Agno (0.665) and OpenAI (0.750), achieving an F1-score of 0.780, while staying ahead of the plain LLM (0.713). For the PROD category, its performance was almost identical to that of the plain LLM, but the Google agent had a very slight edge. Figure 4 compares the F1-scores by entity class.

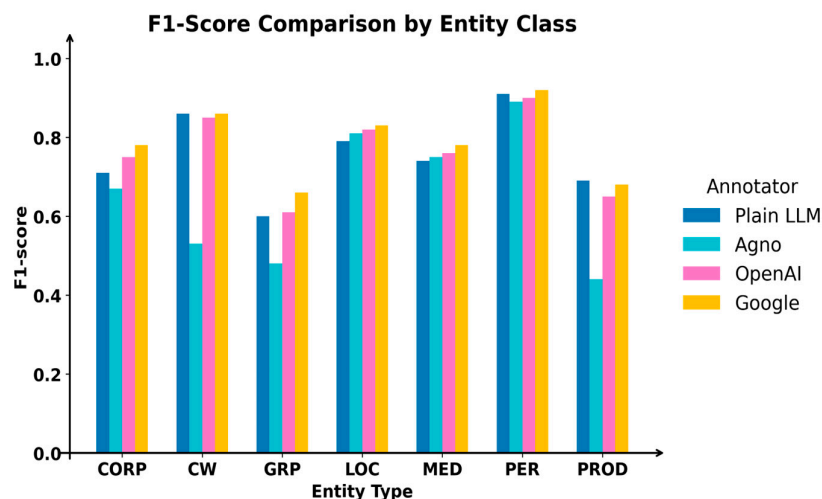


Figure 4. Comparison of annotator performance by entity class.

6. Result Analysis

6.1. Human Annotation

A primary challenge in human annotation is achieving consistency in determining entity boundaries. For example, with the entity سوار هواوي الذكي (“Huawei Smart Band”), some annotators include الذكي (“smart”) as part of the PROD entity, while others classify it as O. Similarly, for نظام التشغيل أي أو إس (“iOS operating system”), some annotators label the entire phrase as a PROD entity, whereas others exclude نظام التشغيل (“operating system”) and label it as O, considering only أي أو إس. Ambiguity also arises with certain entity mentions, such as برنامج أوبن أيه أي, where some annotators label برنامج أوبن أيه أي as CW, while others treat برنامج as O, since there is no actual application called “OpenAI,” and instead label أوبن أيه أي as CORP.

6.2. LLM/Agent Annotation

6.2.1. Hallucination Behavior

Apart from achieving top-level performance, an additional rationale for selecting the Google agent is its lower tendency to hallucinate. During annotation, both the Google agent and plain GPT-4 strictly adhered to the seven NER categories specified in the prompt, without generating any labels beyond those categories. Although the prompt explicitly restricted the task to seven tags, we observed that the Agno and OpenAI agents generated additional, unintended labels. For instance, in addition to the predefined categories (PER, LOC, GRP, CORP, MED, CW, PROD), they generated extraneous labels such as B-O, MISC, BOOK-TITLE, NATIONALITY, TIME, DATE, WORK, AWARD, ORG, and EVENT. In certain cases, the two agents showed inconsistency, labeling the same entity as PER and CW in some instances and as PERSON and CREATIVE_WORK in others. This behavior highlights the robustness of the Google agent, which proved more dependable by remaining aligned with the specified annotation schema.

6.2.2. Error Analysis and Annotator Framework Effects

We conducted an entity-level error analysis on true positives, false positives, and false negatives, categorizing errors into missing entities, spurious entities, boundary-related errors, and incorrect entity types. Under relaxed evaluation, missing entities (e.g., gold entities not predicted) and spurious entities (e.g., predicted entities not present in the gold standard) correspond to relaxed false negatives and relaxed false positives,

respectively. Boundary-related errors (e.g., partial span mismatches) are estimated based on the discrepancy between strict and relaxed entity-level F1-scores, capturing cases where predictions partially overlap gold entities but fail exact span matching, with larger gaps indicating greater boundary mismatches. Incorrect-type errors (e.g., correct spans with incorrect labels) are approximated by the excess of strict false positives over relaxed false positives, reflecting label mismatches on otherwise correct spans.

- Missing-entity errors constitute the most prevalent error type across all annotators, especially for low-salience, context-dependent entities. Many misses occur for MED, PROD, and CW entities, which often rely on domain knowledge and lack explicit lexical markers (e.g., جهاز, دواء, مسلسل, فيلم, رواية), as Arabic lacks capitalization cues to signal named entities. Agents with no internal validation loops—particularly the plain LLM—are more likely to overlook such entities, as there is no reflection or validation mechanisms to reassess recall-sensitive spans, where the errors propagate directly to the output, making performance highly dependent on prompt quality and model determinism.
- Spurious-entity errors arise from overgeneralization, whereby common nouns or descriptive phrases are labeled as named entities based on their perceived semantic role rather than true entity identity. Such errors are most prevalent for PROD, CORP, and GRP types. For example, expressions such as “الإحصائيات البريطانية” function as source-like references, leading to their mislabeling as GRP entities. These errors reflect a precision–recall trade-off, where annotators favor labeling ambiguous, role-bearing spans over leaving them untagged.
- Boundary-related errors largely result from the morphological richness of Arabic and the complexity of its noun phrase structure. Annotators frequently differ on whether to include determiners and modifiers—such as adjectives and explanatory appositives—within the entity span. For example, in “العلامة الإمام شمس الدين ابن القيم الجوزية”, the core PER entity is “شمس الدين ابن القيم الجوزية”, yet the presence of honorific modifiers (العلامة, الإمام) leads models to either extend the span beyond the entity or capture only part of the name. This case demonstrates that while models typically recognize the presence of a named entity, precise boundary delimitation remains difficult when descriptive elements are tightly integrated into the noun phrase. Agents with explicit span validation steps, such as Google ADK, exhibit fewer such discrepancies.
- Incorrect entity type assignment occurs when an entity span is accurately detected but assigned an inappropriate semantic label, revealing limitations in category differentiation rather than entity detection. Real-world entities often occupy overlapping functional and conceptual roles; for instance, brand names and product mentions are frequently labeled interchangeably as CORP or PROD, while medical entities are often misclassified as products. The entity “الجواشا” exemplifies this ambiguity, as it is consistently detected but inconsistently categorized as either a medical concept or a product.

Figure 5 presents the overall error analysis, while Figure 6 details errors by entity type across the four annotators. Although Agno agent interactions execute each step deterministically, effective error containment relies primarily on the careful structuring of annotation and reflection steps, rather than on automatic mitigation by the framework. The Agno agent records the highest error counts across all categories, with significant deficiencies in missing-entity and boundary-shifting errors, indicating limited robustness in span identification. The OpenAI Agents SDK demonstrates effectiveness for structured annotation tasks; specifically, the explicit handoff mechanism strongly influences annotation outcomes. Both the OpenAI agents and the plain LLM display intermediate performance, with boundary error remaining the main source of disagreement with human annotations. This highlights the inherent difficulty of exact span identification in Arabic

NER. In contrast, the Google ADK agent achieves the lowest error rates across all evaluated categories, resulting in more balanced performance and higher annotation quality. This improvement is supported by the orchestration layer that enforces sequential validation and consistency checks.

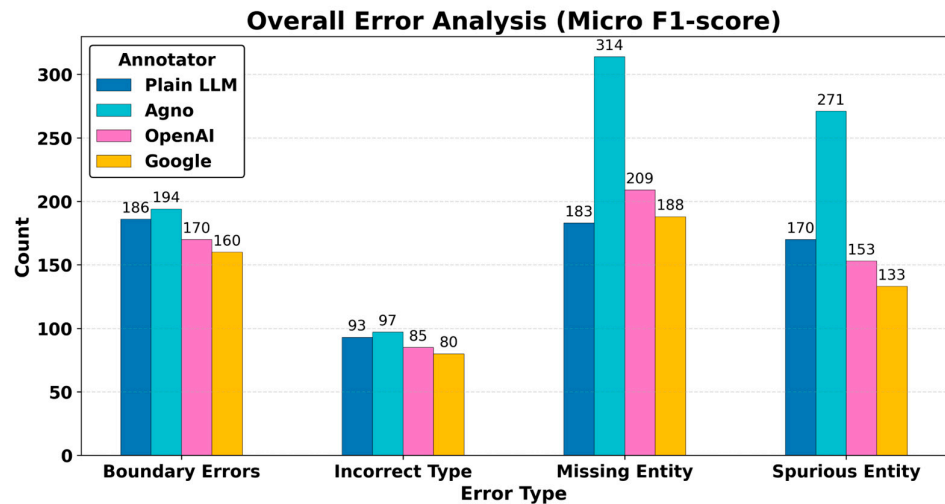
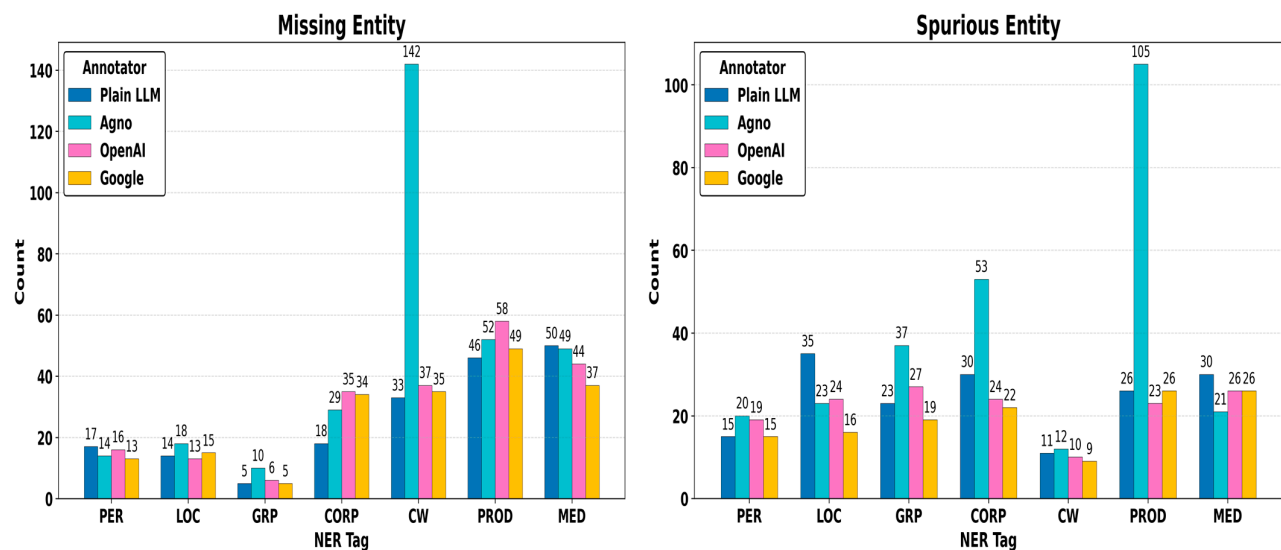


Figure 5. Overall error analysis across annotators.



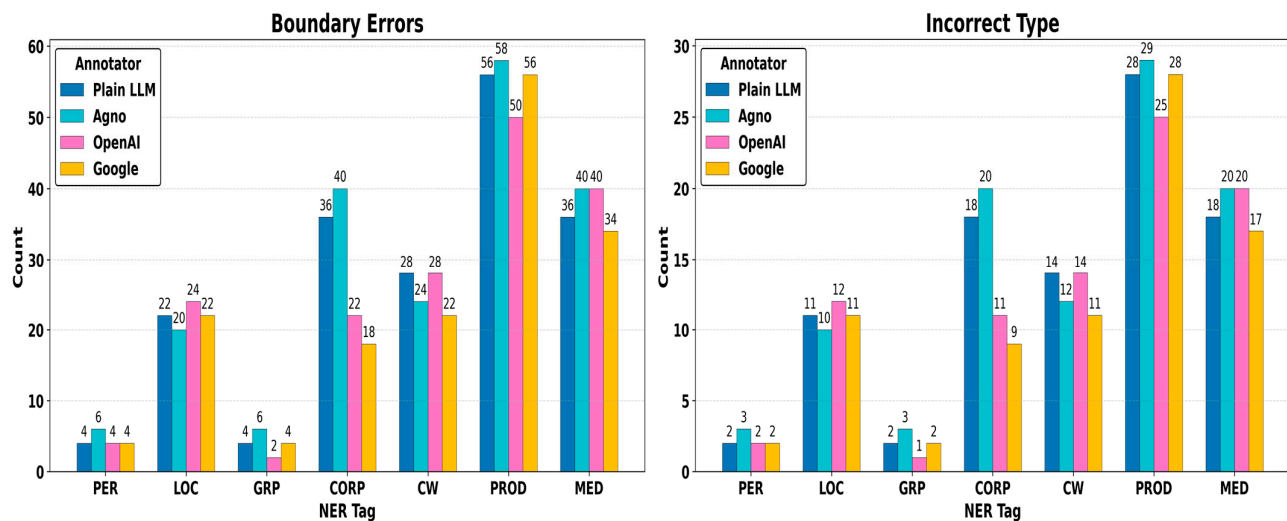


Figure 6. Error analysis by entity type.

6.2.3. MSA vs. Dialectal Arabic

Although evaluation was not explicitly conducted by register, qualitative analysis indicates reliable performance differences between Modern Standard Arabic (MSA) and dialectal Arabic. One major challenge in generating silver labels for Arabic NER is the presence of clitics, which occur more frequently in dialectal Arabic than in MSA, thereby complicating entity annotation. For instance, the entity “كبليل حير□” contains the creative work named entity “كبليل حير□”, but its recognition requires correct segmentation of the proclitic ك (“like”). This is similar to the entity “لخيري بشارة”, which is the person named entity “خيري بشارة” that requires correct segmentation of the proclitic □ (“to”). In MSA, entity names are typically realized in standardized standalone forms with well-defined word boundaries (e.g., الإيطالية, إيطاليا), facilitating accurate span detection and normalization. In contrast, dialectal Arabic commonly attaches prepositions and definite articles to entity names (e.g., للتلائية, بإيطاليا), reducing boundary clarity and leading to missed entities or incorrect normalization.

Compound product naming in spoken contexts presents additional challenges. Dialectal expressions such as “هواتف هواوي ميت” and “هواتف هواوي” are commonly used as general product references, whereas MSA contexts typically use fully specified product names (e.g., هواوي نوفا 10 إس إي). These issues result in systematic CORP–PROD ambiguity, where “هواوي” is annotated as CORP or the full phrase “هواتف هواوي” as PROD in dialectal input.

Overall, these findings support agent-based annotation—particularly the Google ADK agent—over vanilla LLM prompting for Arabic NER in mixed-register settings. Vanilla LLMs remain highly sensitive to morphological attachment and contextual variation because they lack internal validation. In contrast, frameworks with embedded orchestration and consistency checks demonstrate greater robustness and yield more stable annotations across both MSA and dialectal Arabic.

6.3. Single-LLM Limitations

A potential limitation of the proposed framework lies in its reliance on GPT-4-based agents as the primary source of silver annotations. Under this setting, the annotator agent’s systematic annotation preferences or biases may be propagated into the generated dataset and subsequently learned by the task-specific model (TAM). This risk is partially mitigated in our framework by initializing the training process with human-annotated gold data. Therefore, we restricted the LLM’s role to annotation rather than inference and

employed active learning to selectively acquire informative and uncertain samples, rather than uniformly adopt model-generated labels.

6.4. Active Learning Effects

Least-confidence sampling reliably outperforms random sampling across most iterations. While random sampling exhibits fluctuations and unstable performance, least-confidence sampling yields smoother progress and higher F1-scores. In addition, least-confidence sampling shows smoother performance progression and lower variance than random sampling, indicating greater robustness under noisy annotation conditions. Minor declines are observed in later iterations as progressively more complex morphology and ambiguous samples are selected. Such fluctuations are expected behavior in active learning settings, particularly in Arabic NER, which inherently involves annotation uncertainty. Overall, least-confidence sampling still obtains more stable learning and higher peak performance than random sampling. Figure 7 shows the learning curves of the proposed active learning framework as up to 5000 samples are annotated using agent-based annotation.

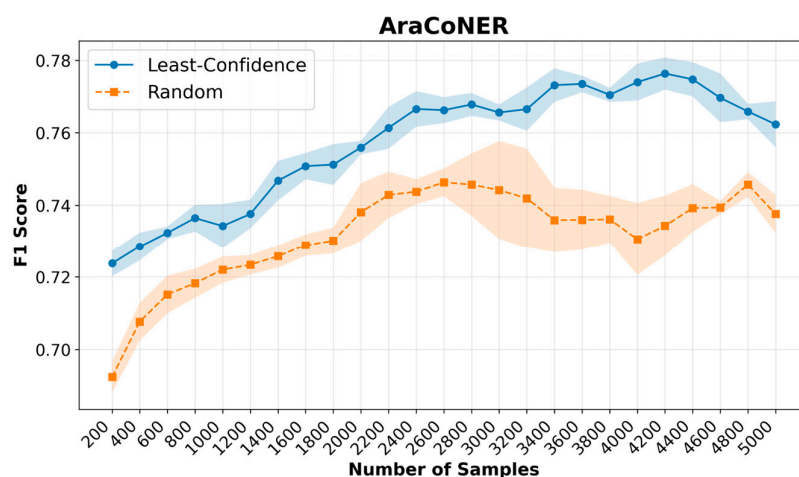


Figure 7. LLMAAA+ performance using least-confidence sampling and random selection over 25 active learning iterations. The plot reports the mean F1-score and standard deviation across multiple random seeds.

7. Overall Results

Table 7 presents the main experimental results obtained on our dataset. LLMAAA+ shows considerable performance gains as the amount of annotated data increases. In particular, scaling LLMAAA+ from 100/500 to 1000/5000 samples improves the average F1-score by approximately 14.3 points ($\approx 23.1\%$ relative improvement). With 1000/5000 samples, LLMAAA+, using least-confidence sampling, outperforms both random sampling and the supervised baseline trained on 1000 clean-labeled instances.

Table 7. Results for LLMAAA+ with different active learning strategies and other baselines. (data: gold/silver), using GPT-4 as the LLM backbone and Google as the agent. The mean F1 $\pm$ standard deviation over three different runs is reported for each method.

Method	#Data	Avg. F1
SUPERVISED	1000/-	0.641 _{0.024}
LLMAAA+ (confidence)	100/500	0.618 _{0.010}
LLMAAA+ (random)	1000/5000	0.738 _{0.006}
LLMAAA+ (confidence)	1000/5000	0.760_{0.011}

Note: #Data denotes the number of instances, reported as gold/silver.

8. Conclusions

In this work, we constructed a corpus of non-traditional names that is rich in new and emerging entities, linguistic trends, and terminology, with minimal human intervention. We propose an Arabic complex NER (AraCoNER) dataset that integrates both human-annotated (gold) and LLM-generated (silver) labels. This is achieved using the proposed LLM-agent-based annotation framework (LLMAAA+), which employs the best-performing LLM-powered agents as active annotators to mitigate annotation costs while maintaining quality and efficiency. Furthermore, LLMAAA+ enables an LLM-powered agent to annotate the most information-rich samples in active learning, incorporate prompt engineering using k-NN example retrieval as in-context examples to train TAM, and then generate pseudo-labels. LLMAAA+ demonstrates superior performance when trained on a combination of gold and silver labels, achieving a higher F1-score compared to training on gold/clean labels exclusively. This approach delivers high performance with good cost and time efficiency, minimizing both annotation expenses and human effort.

As future work, we plan to extend our evaluation by comparing LLMAAA+ with contemporary zero- and few-shot Arabic NER approaches, as well as with other LLM-based annotation frameworks, to further assess its effectiveness and generalizability across different annotation strategies and NER model architectures.

Author Contributions: Conceptualization, W.A., N.A. and I.A.-T.; methodology, W.A., N.A. and I.A.-T.; software, W.A.; validation, W.A., N.A. and I.A.-T.; formal analysis, W.A., N.A. and I.A.-T.; investigation, W.A.; resources, W.A.; data curation, W.A.; writing—original draft preparation, W.A.; writing—review and editing, W.A., N.A. and I.A.-T.; visualization, W.A.; supervision, N.A. and I.A.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the Ongoing Research Funding Program (ORF-2026-857), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: This study was reviewed by the Standing Local Committee for Research Ethics at King Saud University and granted exemption (Ref. No. KSU-HE-25-1223).

Data Availability Statement: Data are available as part of the article and are free for use and redistribution for research purposes. The data are accessible at <https://github.com/AraCoNERI/Dataset> (accessed on 7 January 2026).

Acknowledgments: We appreciate the anonymous reviewers for their valuable comments on this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADK	Agent Development Kit
CSV	Comma-Separated Values
IAA	Inter-Annotator Agreement
k-NN	k-Nearest Neighbor
LLM	Large Language Model
MSA	Modern Standard Arabic
NER	Named Entity Recognition
NLP	Natural Language Processing
TAM	Task Specific Model

Appendix A

Prompts

The complete prompts used for annotation and generation are presented in Table A1. Before finalizing our prompt design, we evaluated it on 50 samples to ensure comprehensive label coverage across all tags. Based on this result, we decided to incorporate additional instructions. To incorporate demonstrations, the text of each sample was added to the input while its label was placed in the output. The target sample was then inserted into the final input, and the corresponding output was left blank for prediction. Table A2 presents the prompts used for generation, which follow a similar design to those for annotation.

Table A1. Annotator prompt.

Description	You are a highly intelligent and accurate news domain named-entity recognition (NER) system in Arabic language.
	Your task is to recognize, and extract named entities in the ‘user input’ and classify them into a set of following predefined entity types: [person (PER), location (LOC), group (GRP), corporation (CORP), Medical (MED), product (PROD), creative work (CW)]
Instruction	Your Output is in JSON format: [{"span": "text", "type": "entity_type"}, ...]
	Scan the user input completely to find all named entity spans. The span must match the exact words in the original text, including white spaces. Ensure no overlapping or nested spans. Spans may include multiple tokens. Only use the allowed entity types: {PER, LOC, CORP, GRP, PROD, CW, MED}. Classification Guidelines:
Output format	• PER: Names of individuals. • LOC: Names of places. • CORP: Names of corporations/business. • GRP: Names of groups/organization. • PROD: Names of products such as clothes, vehicles, food, hardware. • CW: Names of creative works such as movies, books, songs, shows. • MED: Names of diseases, medications, diagnosis.
	Input: "Input: {}" Output: "Output: {}"

Table A2. Generator prompt.

Description	You are an intelligent text data generator. Generate {} high-quality and diverse sentences in news domain containing entities in Arabic language for the following types: [person (PER),
-------------	--

	location (LOC), group (GRP), corporation (CORP), Medical (MED), product (PROD), creative work (CW)]
Format	Example: “Example: {}” Output: “Output: {}”
Default example	{“text”: text, “entities”: [{“name”: name, “type”: type}]}

Appendix B

Annotation Examples

Table B1 shows two sentence queries annotated by the four annotators, illustrating valid and partially valid cases. Table B2 presents a sentence query, k-NN retrieval examples, and silver labels generated by the Google agents.

Table A3. Annotation examples for the four annotators.

Query 1	سحيلة بنت عديم أم الخلاجين من بطولة زهرة الخرجي وحمد العماني وعبد الله التركماني وميس قمر.
Plain GPT-4	[{‘span’: ‘سحيلة بنت عديم’, ‘type’: ‘CW’}, {‘span’: ‘زهرة الخرجي’, ‘type’: ‘PER’}, {‘span’: ‘حمد العماني’, ‘type’: ‘PER’}, {‘span’: ‘عبد الله التركماني’, ‘type’: ‘PER’}, {‘span’: ‘ميس قمر’, ‘type’: ‘PER’}]
GPT-4 + Agno	[{‘span’: ‘سحيلة بنت عديم’, ‘type’: ‘PER’}, {‘span’: ‘زهرة الخرجي’, ‘type’: ‘PER’}, {‘span’: ‘حمد العماني’, ‘type’: ‘PER’}, {‘span’: ‘عبد الله التركماني’, ‘type’: ‘PER’}, {‘span’: ‘ميس قمر’, ‘type’: ‘PER’}]
GPT-4 + OpenAI	[{‘span’: ‘سحيلة بنت عديم’, ‘type’: ‘PER’}, {‘span’: ‘زهرة الخرجي’, ‘type’: ‘PER’}, {‘span’: ‘حمد العماني’, ‘type’: ‘PER’}, {‘span’: ‘عبد الله التركماني’, ‘type’: ‘PER’}, {‘span’: ‘ميس قمر’, ‘type’: ‘PER’}]
GPT-4 + Google	[{‘span’: ‘سحيلة بنت عديم’, ‘type’: ‘CW’}, {‘span’: ‘زهرة الخرجي’, ‘type’: ‘PER’}, {‘span’: ‘حمد العماني’, ‘type’: ‘PER’}, {‘span’: ‘عبد الله التركماني’, ‘type’: ‘PER’}, {‘span’: ‘ميس قمر’, ‘type’: ‘PER’}]
Query 2	واشتهر في الفترة الأخيرة أسياخ شاورما الشيش طاووق الذي يشوى على سيخ عمودي.
Plain GPT-4	[{‘span’: ‘أسياخ شاورما’, ‘type’: ‘PROD’}, {‘span’: ‘الشيش طاووق’, ‘type’: ‘PROD’}]
GPT-4 + Agno	[{‘span’: ‘أسياخ شاورما’, ‘type’: ‘PROD’}, {‘span’: ‘الشيش طاووق’, ‘type’: ‘PROD’}]
GPT-4 + OpenAI	[{‘span’: ‘شاورما’, ‘type’: ‘PROD’}, {‘span’: ‘الشيش طاووق’, ‘type’: ‘PROD’}]
GPT-4 + Google	[{‘span’: ‘شاورما’, ‘type’: ‘PROD’}, {‘span’: ‘الشيش طاووق’, ‘type’: ‘PROD’}]

Table A4. Annotation examples (K-NN and silver labels).

Query sentence	الشركة المطورة لشات جي بي تي تطلق ميزة جديدة لتعزيز منافستها لـ أليكسا وسيري.
k-NN examples	Example 1: Sentence: أعلنت مايكروسوفت إضافة سمات جديدة مدعومة بالذكاء الاصطناعي إلى خدمة البريد الإلكتروني أوت لوك وتتطلب هذه الميزة الاشتراك في كوابلنت برو Entities: كوابلنت برو (CW), أوت لوك (CW), مايكروسوفت (CORP)
	Example 2: Sentence: ذكاء اصطناعي بميزات بشرية. تحديثات استثنائية لأبل في مساعدتها سيري Entities: سيري (CW), لأبل (CORP)
	Example 3: Sentence: بدأت مايكروسوفت اختبار خطة جديدة في خدمات تتضمن مزايا الذكاء الاصطناعي كوابلوت وتطبيق المصمم مع اشتراك أعلى Entities: المصمم (CW), كوابلوت (CW), مايكروسوفت (CORP)
	Example 4: Sentence: كشفت أوبن إيه أي عن محركها البحثي الجديد بالذكاء الاصطناعي سيرش جي بي تي Entities: سيرش جي بي تي (CW), أوبن إيه أي (CORP)
	Example 5: Sentence: تختبر شركة بايت دانس مالكة تطبيق تيك توك تطبيقا جديدا يدعى وهو تطبيق من فئة تطبيقات التواصل الاجتماعي Entities: بايت دانس (CORP), تيك توك (CW)
Silver labels	[{‘span’: ‘شات جي بي تي’, ‘type’: ‘CW’}, {‘span’: ‘أليكسا’, ‘type’: ‘CW’}, {‘span’: ‘سيري’, ‘type’: ‘CW’}]

Appendix C

Table A5. Token-level agreement (human vs. LLM/agent).

Tag	Plain LLM				Agno				OpenAI				Google			
	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1
PER	316	30	32	0.91	313	42	32	0.89	314	37	34	0.90	317	28	29	0.92
LOC	148	47	33	0.79	144	33	34	0.81	150	34	33	0.82	149	28	34	0.83
GRP	56	57	18	0.60	45	70	29	0.48	53	47	21	0.61	55	37	19	0.66
CORP	163	84	47	0.71	169	111	59	0.67	168	50	62	0.75	176	45	54	0.78
CW	377	32	95	0.86	183	35	289	0.53	367	26	105	0.85	378	28	92	0.86
PROD	198	69	114	0.69	165	214	132	0.44	177	52	135	0.65	197	69	115	0.68
MED	275	71	124	0.74	275	58	124	0.75	286	69	113	0.76	291	68	94	0.78
Micro	1533	390	463	0.78	1294	563	699	0.67	1515	315	503	0.79	1563	303	437	0.81

Table A6. Entity-level agreement (relaxed F1-score of human vs. LLM/agent).

Tag	Plain LLM				Agno				OpenAI				Google			
	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1
PER	160	15	17	0.91	162	20	14	0.91	161	19	16	0.90	163	15	13	0.92
LOC	105	35	14	0.81	101	23	18	0.83	108	24	13	0.85	106	16	15	0.82
GRP	30	23	5	0.68	25	37	10	0.52	29	27	6	0.64	30	19	5	0.71
CORP	131	30	18	0.85	137	53	29	0.77	132	24	35	0.82	133	22	34	0.83
CW	207	11	33	0.90	98	12	142	0.56	203	10	37	0.90	204	9	35	0.90
PROD	134	26	46	0.79	121	105	52	0.61	122	23	58	0.75	131	26	49	0.78
MED	141	30	50	0.78	142	21	49	0.80	147	26	44	0.81	148	26	37	0.82
Micro	908	170	183	0.84	786	271	314	0.73	902	153	209	0.83	915	133	188	0.85

Table A7. Entity-level agreement (strict F1-score of human vs. LLM/agent).

Tag	Plain LLM				Agno				OpenAI				Google			
	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1	TP	FP	FN	F1
PER	158	17	19	0.90	159	23	17	0.89	159	21	18	0.90	161	17	15	0.91
LOC	94	46	25	0.73	91	33	28	0.75	96	36	25	0.76	95	27	26	0.78
GRP	28	25	7	0.64	22	40	13	0.45	28	28	7	0.62	28	21	7	0.67
CORP	113	48	36	0.73	117	73	49	0.66	121	35	46	0.75	124	31	43	0.77
CW	193	25	47	0.84	86	24	154	0.50	189	24	51	0.83	193	20	46	0.85
PROD	106	54	74	0.62	92	134	81	0.46	97	48	83	0.60	103	54	77	0.61
MED	123	48	68	0.68	122	41	69	0.69	127	46	64	0.70	131	43	54	0.73
Micro	815	263	276	0.75	689	368	411	0.64	817	238	294	0.75	835	213	268	0.78

Appendix D

Implementation

All experiments were conducted with three random seeds, yielding different parameter initializations and data samples, and the results are reported as mean and standard deviation. Sentence embeddings were computed using Arabic-SBERT-100K and utilized for KNN-based retrieval with cosine similarity and $k = 5$ neighbors to select few-shot examples. AraBERT was adopted as the encoder for TAMs, trained with a learning rate of 5×10^{-5} , and optimized using AdamW.

References

1. Khodabakhsh, M.; Kahani, M.; Bagheri, E.; Noorian, Z. Detecting Life Events from Twitter Based on Temporal Semantic Features. *Knowl.-Based Syst.* **2018**, *148*, 1–16. <https://doi.org/10.1016/j.knosys.2018.02.021>.
2. Gupta, P.; Gupta, V. A Survey of Text Question Answering Techniques. *Int. J. Comput. Appl.* **2012**, *53*, 1–8. <https://doi.org/10.5120/8406-2030>.
3. Qiang, J.; Chen, P.; Ding, W.; Wang, T.; Xie, F.; Wu, X. Heterogeneous-Length Text Topic Modeling for Reader-Aware Multi-Document Summarization. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 1–21. <https://doi.org/10.1145/3333030>.
4. Guo, J.; Xu, G.; Cheng, X.; Li, H. Named Entity Recognition in Query. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, 19–23 July 2009; pp. 267–274.
5. Boldi, P.; Bonchi, F.; Castillo, C.; Donato, D.; Gionis, A.; Vigna, S. The Query-Flow Graph: Model and Applications. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; ACM: New York, NY, USA, 2008; pp. 609–618.
6. Ashwini, S.; Choi, J.D. Targetable Named Entity Recognition in Social Media. *arXiv* **2014**, arXiv:1408.0782.
7. Chen, B.; Ma, J.-Y.; Qi, J.; Guo, W.; Ling, Z.-H.; Liu, Q. USTC-NELSLIP at SemEval-2022 Task 11: Gazetteer-Adapted Integration Network for Multilingual Complex Named Entity Recognition. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Online, 14–15 July 2022; Association for Computational Linguistics: Strassburg, PA, USA, 2022; pp. 1613–1622.
8. Meng, T.; Fang, A.; Rokhlenko, O.; Malmasi, S. GEMNET: Effective Gated Gazetteer Representations for Recognizing Complex Entities in Low-Context Input. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; Association for Computational Linguistics: Strassburg, PA, USA, 2021; pp. 1499–1512.
9. Malmasi, S.; Fang, A.; Fetahu, B.; Kar, S.; Rokhlenko, O. MultiCoNER: A Large-Scale Multilingual Dataset for Complex Named Entity Recognition. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; International Committee on Computational Linguistics: Gyeongju, Republic of Korea; pp. 3798–3809.
10. Bogdanov, S.; Constantin, A.; Bernard, T.; Crabbé, B.; Bernard, E.P. NuNER: Entity Recognition Encoder Pre-Training via LLM-Annotated Data. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; Association for Computational Linguistics: Strassburg, PA, USA, 2024; pp. 11829–11841.
11. Salah, R.E.; Zakaria, L.Q.B. Building the Classical Arabic Named Entity Recognition Corpus (CANERCorpus). In Proceedings of the 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), Kota Kinabalu, Malaysia, 26–28 March 2018; pp. 1–8.
12. Mohit, B.; Schneider, N.; Bhowmick, R.; Oflazer, K.; Smith, N.A. Recall-Oriented Learning of Named Entities in Arabic Wikipedia. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, 23–27 April 2012; Association for Computational Linguistics: Strassburg, PA, USA, 2012; pp. 162–173.
13. Fetahu, B.; Chen, Z.; Kar, S.; Rokhlenko, O.; Malmasi, S. MultiCoNER v2: A Large Multilingual Dataset for Fine-Grained and Noisy Named Entity Recognition. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; Association for Computational Linguistics: Strassburg, PA, USA, 2023; pp. 2027–2051.
14. Miftahova, A.; Pugachev, A.; Skiba, A.; Artemova, E.; Batura, T.; Braslavski, P. NamedEntityRangers at SemEval-2022 Task 11: Transformer-Based Approaches for Multilingual Complex Named Entity Recognition. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Online, 14–15 July 2022; Association for Computational Linguistics: Strassburg, PA, USA, 2022; pp. 1570–1575.
15. Çarık, B.; Beyhan, F.; Yeniterzi, R. SU-NLP at SemEval-2022 Task 11: Complex Named Entity Recognition with Entity Linking. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Online, 14–15 July 2022; Association for Computational Linguistics: Strassburg, PA, USA, 2022; pp. 1648–1653.
16. Pan, J.Z.; Razniewski, S.; Kalo, J.-C.; Singhanian, S.; Chen, J.; Dietze, S.; Jabeen, H.; Omeliyanenko, J.; Zhang, W.; Lissandrini, M.; et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. *arXiv* **2023**, arXiv:2308.06374.
17. Kommineni, V.K.; König-Ries, B.; Samuel, S. From Human Experts to Machines: An LLM Supported Approach to Ontology and Knowledge Graph Construction. *arXiv* **2024**, arXiv:2403.08345.
18. Benajiba, Y.; Rosso, P.; BenedíRuiz, J.M. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In *Computational Linguistics and Intelligent Text Processing*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4394, pp. 143–153.

19. Xue, N.; Palmer, M.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; Belvin, R.; Houston, A. OntoNotes Release 4.0. In Proceedings of the Linguistic Data Consortium, Philadelphia, PA, USA, 17 February 2011.
20. Jarrar, M.; Khalilia, M.; Ghanem, S. Wojood: Nested Arabic Named Entity Corpus and Recognition Using BERT. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; European Language Resources Association: Paris, France, 2022; pp. 3626–3636.
21. Jarrar, M.; Hamad, N.; Khalilia, M.; Talafha, B.; Elmadany, A.; Abdul-Mageed, M. WojoodNER 2024: The Second Arabic Named Entity Recognition Shared Task. In Proceedings of the Second Arabic Natural Language Processing Conference, Bangkok, Thailand, 16 August 2024; Association for Computational Linguistics: Strassburg, PA, USA, 2024; pp. 847–857.
22. Foppiano, L.; Lambard, G.; Amagasa, T.; Ishii, M. Mining Experimental Data from Materials Science Literature with Large Language Models: An Evaluation Study. *Sci. Technol. Adv. Mater. Methods* **2024**, *4*, 2356506. <https://doi.org/10.1080/27660400.2024.2356506>.
23. Hou, W.; Zhao, W.; Liu, X.; Guo, W. Knowledge-Enriched Prompt for Low-Resource Named Entity Recognition. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2024**, *23*, 72:1–72:15. <https://doi.org/10.1145/3659948>.
24. Li, M.; Zhou, H.; Yang, H.; Zhang, R. RT: A Retrieving and Chain-of-Thought Framework for Few-Shot Medical Named Entity Recognition. *J. Am. Med. Inform. Assoc.* **2024**, *31*, 1929–1938. <https://doi.org/10.1093/jamia/ocae095>.
25. Zhou, W.; Zhang, S.; Gu, Y.; Chen, M.; Poon, H. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition *arXiv* **2023**, arXiv:2308.03279.
26. Wang, S.; Sun, X.; Li, X.; Ouyang, R.; Wu, F.; Zhang, T.; Li, J.; Wang, G.; Guo, C. GPT-NER: Named Entity Recognition via Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, 29 April–4 May 2025; Association for Computational Linguistics: Strassburg, PA, USA, 2025; pp. 4257–4275.
27. Xie, T.; Li, Q.; Zhang, Y.; Liu, Z.; Wang, H. Self-Improving for Zero-Shot Named Entity Recognition with Large Language Models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), Mexico City, Mexico, 16–21 June 2024; Duh, K., Gomez, H., Bethard, S., Eds.; Association for Computational Linguistics: Strassburg, PA, USA, 2024; pp. 583–593.
28. Ashok, D.; Lipton, Z.C. PromptNER: Prompting For Named Entity Recognition. *arXiv* **2023**, arXiv:2305.15444.
29. Hu, Y.; Chen, Q.; Du, J.; Peng, X.; Keloth, V.K.; Zuo, X.; Zhou, Y.; Li, Z.; Jiang, X.; Lu, Z.; et al. Improving Large Language Models for Clinical Named Entity Recognition via Prompt Engineering. *J. Am. Med. Inform. Assoc.* **2024**, *31*, 1812–1820. <https://doi.org/10.1093/jamia/ocad259>.
30. Zhu, X.; Dai, F.; Gu, X.; Li, B.; Zhang, M.; Wang, W. *GL-NER: Generation-Aware Large Language Models for Few-Shot Named Entity Recognition*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 433–448.
31. Zhang, X.; Chen, G.; Cui, S.; Sheng, J.; Liu, T.; Xu, H. Exogenous and Endogenous Data Augmentation for Low-Resource Complex Named Entity Recognition. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; ACM: New York, NY, USA, 2024; pp. 630–640.
32. Zhu, Y.; Zhang, P.; Haq, E.-U.; Hui, P.; Tyson, G. Can ChatGPT Reproduce Human-Generated Labels? A Study of Social Computing Tasks. *arXiv* **2023**, arXiv:2304.10145.
33. Törnberg, P. ChatGPT-4 Outperforms Experts and Crowd Workers in Annotating Political Twitter Messages with Zero-Shot Learning *arXiv* **2023**, arXiv:2304.06588.
34. Ding, B.; Qin, C.; Liu, L.; Chia, Y.K.; Li, B.; Joty, S.; Bing, L. Is GPT-3 a Good Data Annotator? In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; Association for Computational Linguistics: Strassburg, PA, USA, 2023; pp. 11173–11195.
35. Wang, S.; Liu, Y.; Xu, Y.; Zhu, C.; Zeng, M. Want To Reduce Labeling Cost? GPT-3 Can Help. In Proceedings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; Moens, M.-F., Huang, X., Specia, L., Yih, S.W., Eds.; Association for Computational Linguistics: Strassburg, PA, USA, 2021; pp. 4195–4205.
36. Meoni, S.; Ryffel, T.; Villemonte de La Clergerie, E. Large Language Models as Instructors: A Study on Multilingual Clinical Entity Extraction. In Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, Toronto, ON, Canada, 13 July 2023; Association for Computational Linguistics: Strassburg, PA, USA, 2023; pp. 178–190.
37. Gilardi, F.; Alizadeh, M.; Kubli, M. ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks. *Proc. Natl. Acad. Sci. USA* **2023**, *120*, e2305016120.
38. Alizadeh, M.; Kubli, M.; Samei, Z.; Dehghani, S.; Zahedivafa, M.; Bermeo, J.; Korobeynikova, M.; Gilardi, F. Open-Source LLMs for Text Annotation: A Practical Guide for Model Setting and Fine-Tuning. *J. Comput. Soc. Sci.* **2025**, *8*, 17.

39. Li, L.; Fan, L.; Atreja, S.; Hemphill, L. “HOT” ChatGPT: The Promise of ChatGPT in Detecting and Discriminating Hateful, Offensive, and Toxic Comments on Social Media. *ACM Trans. Web* **2024**, *18*, 30:1–30:36. <https://doi.org/10.1145/3643829>.
40. Nasution, A.H.; Onan, A. ChatGPT Label: Comparing the Quality of Human-Generated and LLM-Generated Annotations in Low-Resource Language NLP Tasks. *IEEE Access* **2024**, *12*, 71876–71900.
41. Chen, R.; Qin, C.; Jiang, W.; Choi, D. Is a Large Language Model a Good Annotator for Event Extraction? *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 17772–17780. <https://doi.org/10.1609/aaai.v38i16.29730>.
42. Mizrahi, M.; Kaplan, G.; Malkin, D.; Dror, R.; Shahaf, D.; Stanovsky, G. State of What Art? A Call for Multi-Prompt LLM Evaluation. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 933–949.
43. Zhu, K.; Wang, J.; Zhou, J.; Wang, Z.; Chen, H.; Wang, Y.; Yang, L.; Ye, W.; Zhang, Y.; Gong, N.; et al. PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. In Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis, Salt Lake City, UT, USA, 14–18 November 2023; ACM: New York, NY, USA, 2023; pp. 57–68.
44. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-Y.; Li, Z.; Gupta, B.B.; Chen, X.; Wang, X. A Survey of Deep Active Learning. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–40.
45. Zhan, X.; Wang, Q.; Huang, K.; Xiong, H.; Dou, D.; Chan, A.B. A Comparative Survey of Deep Active Learning. *arXiv* **2022**, arXiv:2203.13450.
46. Diao, S.; Wang, P.; Lin, Y.; Pan, R.; Liu, X.; Zhang, T. Active Prompting with Chain-of-Thought for Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 11–16 August 2024; Association for Computational Linguistics: Strausburg, PA, USA, 2024; pp. 1330–1350.
47. *ActiveLLM: Large Language Model-Based Active Learning for Textual Few-Shot Scenarios*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 89–112.
48. Bhatt, G.; Chen, Y.; Das, A.; Zhang, J.; Truong, S.; Musmann, S.; Zhu, Y.; Bilmes, J.; Du, S.; Jamieson, K.; et al. An Experimental Design Framework for Label-Efficient Supervised Finetuning of Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024; Association for Computational Linguistics: Strausburg, PA, USA, 2024; pp. 6549–6560.
49. Yang, R.; Pan, X.; Luo, F.; Qiu, S.; Zhong, H.; Yu, D.; Chen, J. Rewards-in-Context: Multi-Objective Alignment of Foundation Models with Dynamic Preference Adjustment. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; JMLR.org: Brookline, MA, USA, 2024; Volume 235, pp. 56276–56297.
50. Wang, X.; Kim, H.; Rahman, S.; Mitra, K.; Miao, Z. Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–21.
51. Lu, Y.; Yao, B.; Zhang, S.; Wang, Y.; Zhang, P.; Lu, T.; Li, T.J.-J.; Wang, D. Human Still Wins over LLM: An Empirical Study of Active Learning on Domain-Specific Annotation Tasks. *arXiv* **2023**, arXiv:2311.09825.
52. Fernandes, P.; Madaan, A.; Liu, E.; Farinhas, A.; Martins, P.H.; Bertsch, A.; De Souza, J.G.C.; Zhou, S.; Wu, T.; Neubig, G.; et al. Bridging the Gap: A Survey on Integrating (Human) Feedback for Natural Language Generation. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 1643–1668. https://doi.org/10.1162/tacl_a_00626.
53. Derczynski, L.; Nichols, E.; van Erp, M.; Limsopatham, N. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In Proceedings of the 3rd Workshop on Noisy User-generated Text; Association for Computational Linguistics, Copenhagen, Denmark, 7 September 2017; pp. 140–147.
54. Eid, Y.; Zayed, H.; Medhat, W. A-MASA: Arabic Multi-Domain Aspect-Based Sentiment Analysis Datasets. *Procedia Comput. Sci.* **2024**, *244*, 202–211. <https://doi.org/10.1016/j.procs.2024.10.193>.
55. Aly, M.; Atiya, A. LABR: A Large Scale Arabic Book Reviews Dataset. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Sofia, Bulgaria, 4–9 August 2013; Association for Computational Linguistics: Strausburg, PA, USA, 2013; pp. 494–498.
56. Tjong Kim Sang, E.F.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL, Edmonton, AB, Canada, 31 May–1 June 2003; pp. 142–147.
57. LDC. *Livable*, Version 5.6.1 2005.05.23; ACE (Automatic Content Extraction) English Annotation Guidelines for Entities; Linguistic Data Consortium: Philadelphia, PA, USA, 2004.
58. Rizzo, G.; van Erp, M. Making Sense of Microposts (#Microposts2016) Named Entity Recognition and Linking (NEEL) Challenge. In Proceedings of the Microposts2016, Montréal, QC, Canada, 11 April 2016; pp. 44–53.

59. Khandelwal, A.; Kar, A.; Chikka, V.R.; Karlapalem, K. Biomedical NER Using Novel Schema and Distant Supervision. In Proceedings of the Proceedings of the 21st Workshop on Biomedical Language Processing; Association for Computational Linguistics, Dublin, Ireland, 26 May 2022; pp. 155–160.
60. Dai, X.; Erp, M. van; Plu, J.; Troncy, R. Recognizing Complex Entity Mentions: A Review and Future Directions. In Proceedings of the ACL 2018, Student Research Workshop; Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 37–44.
61. Schneider, N.; Danchik, E.; Dyer, C.; Smith, N.A. Discriminative Lexical Semantic Segmentation with Gaps: Running the MWE Gamut. *Trans. Assoc. Comput. Linguist.* **2014**, *2*, 193–206. https://doi.org/10.1162/tacl_a_00176.
62. Hripcsak, G.; Rothschild, A.S. Agreement, the F-Measure, and Reliability in Information Retrieval | Journal of the American Medical Informatics Association. *J. Am. Med. Inform. Assoc.* **2005**, *12*, 296–298.
63. Campillos-Llanos, L.; Valverde-Mateos, A.; Capllonch-Carrión, A.; Moreno-Sandoval, A. A Clinical Trials Corpus Annotated with UMLS Entities to Enhance the Access to Evidence-Based Medicine. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 69.
64. Brandsen, A.; Verberne, S.; Wansleebe, M.; Lambers, K. Creating a Dataset for Named Entity Recognition in the Archaeology Domain. In Proceedings of the Twelfth Language Resources and Evaluation Conference; European Language Resources Association, Marseille, France, 11–16 May 2020; pp. 4573–4577.
65. Ogren, P.; Savova, G.; Chute, C. Constructing Evaluation Corpora for Automated Clinical Named Entity Recognition. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, 28–30 May 2008; European Language Resources Association (ELRA): Paris, France, 2008.
66. Zhang, R.; Li, Y.; Ma, Y.; Zhou, M.; Zou, L. LLMaAA: Making Large Language Models as Active Annotators. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; Association for Computational Linguistics: Strassburg, PA, USA; pp. 13088–13103.
67. Kuzman, T.; Mozetič, I.; Ljubešić, N. ChatGPT: Beginning of an End of Manual Linguistic Data Annotation? Use Case of Automatic Genre Identification. *arXiv* **2023**, arXiv:2303.03953.
68. Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; Chen, W. What Makes Good In-Context Examples for GPT-3? In Proceedings of the Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Dublin, Ireland, 27 May 2022; Association for Computational Linguistics: Strassburg, PA, USA, 2022; pp. 100–114.
69. Culotta, A.; McCallum, A. *Reducing Labeling Effort for Structured Prediction Tasks*; Defense Technical Information Center: Fort Belvoir, VA, USA, 2005.
70. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-Based Model for Arabic Language Understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 11–16 May 2020; European Language Resource Association: Paris, France, 2020; pp. 9–15.
71. Zhang, X.; Talukdar, N.; Vemulapalli, S.; Ahn, S.; Wang, J.; Meng, H.; Bin Murtaza, S.M.; Leshchiner, D.; Dave, A.A.; Joseph, D.F.; et al. Comparison of Prompt Engineering and Fine-Tuning Strategies in Large Language Models in the Classification of Clinical Notes. *AMIA Summits Transl. Sci. Proc.* **2024**, *2024*, 478–487.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.