Trusted Knowledge Extraction for Operations and Maintenance Intelligence

Kathleen P. Mealey[1], Jonathan A. Karr Jr.[1], Priscila Saboia Moreira[1], Paul R. Brenner[1], Charles F. Vardeman II[11]

[1]University of Notre Dame, Notre Dame, Indiana, USA
kpmealey@outlook.com, {jkarr, pmoreira, paul.r.brenner, cvardema}@nd.edu

Abstract

Deriving operational intelligence from organizational data repositories is a key challenge due to the dichotomy of data confidentiality vs data integration objectives, as well as the limitations of Natural Language Processing (NLP) tools relative to the specific knowledge structure of domains such as operations and maintenance. In this work, we discuss Knowledge Graph construction and break down the Knowledge Extraction process into its Named Entity Recognition, Coreference Resolution, Named Entity Linking, and Relation Extraction functional components. We then evaluate sixteen NLP tools in concert with or in comparison to the rapidly advancing capabilities of Large Language Models (LLMs). We focus on the operational and maintenance intelligence use case for trusted applications in the aircraft industry. A baseline dataset is derived from a rich public domain US Federal Aviation Administration dataset focused on equipment failures or maintenance requirements. We assess the zero-shot performance of NLP and LLM tools that can be operated within a controlled, confidential environment (no data is sent to third parties). Based on our observation of significant performance limitations, we discuss the challenges related to trusted NLP and LLM tools as well as their Technical Readiness Level for wider use in mission-critical industries such as aviation. We conclude with recommendations to enhance trust and provide our open-source curated dataset to support further baseline testing and evaluation.

Keywords: Knowledge Extraction; Knowledge Graphs; Maintenance; Zero-shot

Link to our dataset: https://zenodo.org/records/13333825

Contents

1 Introduction

Organizations in domains such as aviation, manufacturing, and defense generate vast amounts of unstructured data in the form of reports, operational logs, and incident records. These databases hold key insights that can be leveraged to enhance safety procedures, predict maintenance timelines, streamline operations, and more. However, accessing and modeling such insights is challenging. Trends in these operational records, which we dub "operations and maintenance intelligence," are fragmented among thousands of disconnected reports. The reports are often inconsistently structured, and meaningful knowledge is often obscured by industry shorthand and lack of context.

One process with great potential to harness insights in large databases is Knowledge Extraction (KE), in which targeted data points are extracted and captured in a structured form, such as a Knowledge Graph (KG). In a KG, individual data entities are represented as nodes, with edges capturing the semantic relationships between them. Structured data is key for providing operations and maintenance intelligence because it is much more readily searched, analyzed, and verified than unstructured text. There are many effective open-source KE tools available out-of-the-box; however, they are trained on open-domain, conventional prose, and therefore struggle to adapt to the strange vocabulary and syntax used in operations and maintenance records. Organizations require KE tools that are both effective and trustworthy in this context, where trust includes the ability to process their data collections at acceptable levels of accuracy, understandability, robustness, reproducibility, and confidentiality.

Existing research in trustworthy KE has produced robust capabilities in several Natural Language Processing (NLP) techniques. NLP-based methods include Named Entity Recognition (NER), which identifies named entities in text and classifies them into a set of entity types Sundheim (1995). In this study, we adopt a multi-stage KE workflow, based on the Information Extraction Pipeline (Bratanic (2021)), which consists of four core NLP tasks: Coreference Resolution (CR), which links different expressions referring to the same entity and has been shown to increase accuracy in many NLP tasks (Sukthanker et al. (2020)); Named Entity Linking (NEL), which enriches identified entities by linking them to unique identities in external knowledge bases[1]; and Relation Extraction (RE), which identifies meaningful relationships between entities. NER is the fourth task, which is often embedded in NEL and RE, since many NEL and RE systems utilize a multi-stage approach using an NER sub-module to extract entities before linking or relating them, respectively. Together, these steps support the construction of knowledge graphs. The diagram in Figure 1 summarizes the KE Workflow.

These and other NLP-based methods have been proven on several open-domain benchmark datasets. However, applying these KE capabilities to the operations and maintenance domain remains underexplored in open-source lit-

[1]NER is often conflated with Named Entity Disambiguation (NED), which is the task of disambiguating an entity from its possible references, without necessarily linking it to an external KB (Al-Moslmi et al. (2020)).

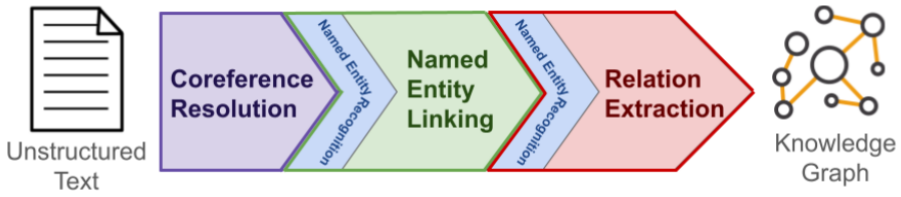

Figure 1: KE Workflow. The Knowledge Extraction Workflow is an approach to extracting graphical data from unstructured text. It begins with CR, which identifies different words or phrases that refer to the same entity. Then, in NEL, entities are recognized (NER) and linked to corresponding unique IDs in an external KB. Lastly, in RE, entities are recognized (NER) and connected through well-defined relationships.

erature. Much of the work in this area is constrained by proprietary datasets, limiting reproducibility and public evaluation. As a result, research efforts are often siloed across organizations with differing data standards, workflows, and infrastructure—making it difficult to compare methods or build on shared benchmarks. A few initiatives, such as MaintNet Akhbardeh et al. (2020a), have begun to address this gap by releasing annotated datasets in the aviation and automotive maintenance domains. However, such resources remain limited in scope, and there is still a need for comprehensive benchmarks to evaluate tool performance in real-world maintenance settings.

As organizations increasingly rely on data-driven approaches for decision support, the inability to extract accurate, structured knowledge from unstructured records buries potentially critical insights. In the maintenance and operations domain, these insights could build systems for safety assurance, performance monitoring, and predictive maintenance, and more. Without trusted KE tools tailored to their specific domain, these organizations face greater operational risk, increased manual overhead, and missed opportunities for insight-driven optimization.

To address this gap, we introduce the Operations and Maintenance Intelligence benchmark, or OMIn, a novel benchmark dataset in the operations and maintenance domain. OMIn is based on curated records from the publicly available FAA Accident/Incident datasets, which shares several peculiarities found in maintenance data: prevalence of rare entities, uncommon or incorrect syntax due to shorthand, abbreviations, acronyms, and small record size. We also release gold standard annotations for NER, CR, and NEL based on OMIn. Note that we did not create a gold standard for RE, for reasons discussed in Section 4.3.4.

We then used OMIn to benchmark sixteen openly available NLP tools on the operations and maintenance domain in a zero-shot setting. While the term zero-shot is used in multiple ways in the field of machine learning, for the purpose of our paper, evaluation in a zero-shot setting means that none of the tools chosen

have been fine-tuned on FAA data nor have been trained for the aviation or maintenance domains. We present our results to inform the selection of off-the-shelf models and identify candidates for domain adaptation or fine-tuning to meet operational requirements.

The diagram in Figure 2 demonstrates the KE workflow performed on a sample from OMIn.

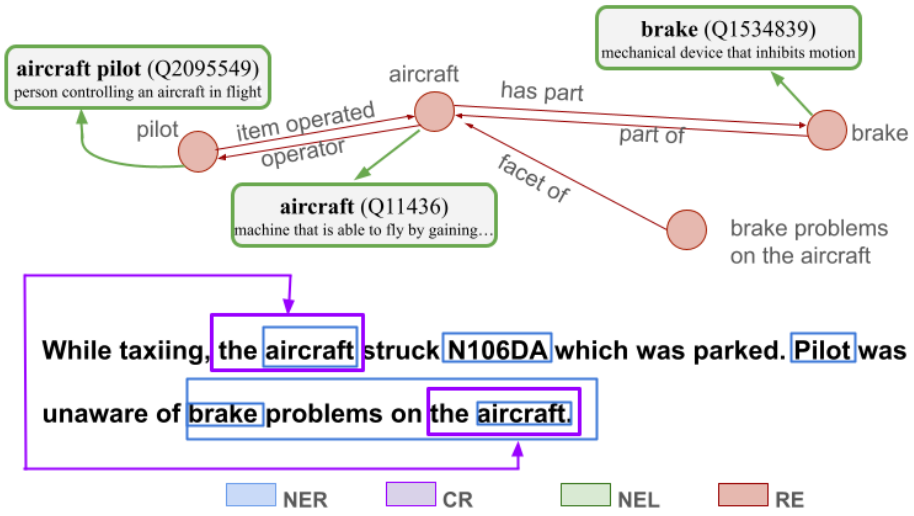

Figure 2: KE Workflow Applied Example. Here, the KE Workflow is applied to the sentence on the bottom from OMIn to generate the graph on the top. Named entities, like *aircraft* and *Pilot*, are denoted in blue to signify the NER subtask in NEL and RE. Then, the CR system (purple) recognizes that the *aircraft* refers to the same entity in different parts of the sentence, ensuring information relating to the aircraft is consolidated around one node. NEL (green) connects recognized entities to their corresponding Wikidata entries, such as *aircraft* (Q11436) and *aircraft pilot* (Q2095549). Finally, RE (red) identifies relationships between entities, with the red edges representing Wikidata properties, such as the pilot *operating* the aircraft or the brake being a *part of* the aircraft.

The key contributions of this work are threefold:

- A publicly released benchmark dataset (OMIn) with gold standards for KE in the maintenance domain.
- A comprehensive zero-shot evaluation of sixteen KE tools using OMIn.
- An analysis of tool performance, limitations, and implications for trusted decision support in maintenance operations.

While we focus on maintenance, this technology has use-cases in many fields, such as healthcare, law, and logistics, where it is important to sort through massive amounts of unstructured data and identify patterns of interest.

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 presents the methodology used in this study, starting with the creation of the OMIn dataset, including gold standards for annotation, then KE tools selection and zero-shot implementation, and finally our evaluation methodology. Section 4 presents the results from evaluating the selected tools on the OMIn dataset. Section 5 discusses the performance, trustworthiness, and technological readiness of the surveyed tools for the maintenance domain based on the results in Section 4. We conclude by summarizing key contributions and listing suggestions for future work.

2 Related Work

The integration of LLMs and KGs is becoming a key area of innovation (Khorashadizadeh et al. (2024)). Together, they create potential for powerful synergy because, while LLMs excel at tasks which require a human-like ability to reason and respond, they often struggle to recall specific facts from their training data. KGs, in turn, may be used to capture such facts – representing knowledge for LLMs to retrieve and for humans to verify and update. KE, a crucial step in creating and updating KGs, is particularly relevant for specialized domains where precise information recall and reasoning are paramount. Given the often fragmented and domain-specific nature of data in many critical industries, effective KE is essential. This integration and its applications lead us to three main points: (1) LLM-KG Advancements for Question-Answering, (2) KE in Maintenance and Aviation Domains, and (3) Open Data and Collaboration in Technical Domains.

2.1 LLM-KG Advancements for Question-Answering

The field of LLM-KG integration and question-answering has seen several advancements. Graph RAG offers an approach to question-answering by populating a KG with facts from text for consistent querying. This method has been shown to outperform Naïve RAG in terms of comprehensiveness, diversity, and empowerment, all while using fewer tokens, ultimately providing better contextual understanding and query scalability (Edge et al. (2024)). Furthermore, Ontology-Based Information Extraction (OBIE) has emerged as a subfield that integrates ontologies into the Information Extraction (IE) process to formally specify which concepts to extract, with existing surveys providing a taxonomy of state-of-the-art OBIE systems (Konys (2018)). Recent work also explores how LLMs can be directly fine-tuned to extract structured information from text and populate KGs, often leveraging prompt engineering and few-shot learning to guide the extraction process for specific schema types (Liu and Li (2025)). This enables more flexible and adaptable KG construction from diverse text

sources.

It is also important that these techniques work with sensitive data. The Llamdex framework (Large LAnguage Model with Domain EXpert) utilizes a private model for domain knowledge (Wu et al. (2024)). Through their secure transfer generation, they create accurate responses to domain-specific questions while preserving KE. During this process, it is important to focus on secure LLM deployment strategies, including: on-premises LLMs, secure RAG, sandboxing, data anonymization, PII scrubbing, differential privacy, access control, encryption, logging, and red-teaming strategies (Matviishyn (2025)). Privacy is important since there can be challenges with LLM training data privacy, insecure user prompts, vulnerabilities with LLM-generated outputs, and issues with LLM agents (Shanmugarasa et al. (2025)).

2.2 KE in Maintenance and Aviation Domains

KE in the maintenance and aviation domains is critical, as large amounts of unstructured textual data often contain vital insights. Various methods have been explored for the construction of KGs reguarding the representation of domain knowledge (Mishra et al. (2017); Yu et al. (2020)). Approaches to maintenance KE include rules-based methods for extracting entities and relations, which can then be augmented by LLMs for additional contextual extraction (Dixit et al. (2021)). The issue of obtaining maintenance information has been framed as a key to finding the root cause of failures, with proposed methods for classifying maintenance records based on latent semantic analysis and SVM (Sharp et al. (2017)). User-friendly tools such as KNOWO have been developed to help create KGs from maintenance work orders, featuring the automatic extraction of concepts belonging to controlled vocabularies (Ameri and Tahsin (2022)). Additionally, KGs have been constructed using datasets of aircraft maintenance information, focusing on components, fault information, and maintenance measures (Yue et al. (2022)). The complexity of aviation maintenance records often contain technical jargon, abbreviations, and informal language, presents unique challenges that domain-specific NLP models and specialized ontologies are increasingly addressing to improve extraction accuracy (Liu et al. (2025)).

2.3 Open Data and Collaboration in Technical Domains

Maintainers and analysts have been proposed to collaborate towards developing standards for textual analysis, including entity typing, which could form the basis for technical language processing (TLP) Brundage et al. (2021). Initiatives like MaintNet provide collections of annotated logbook datasets across aviation, automotive, and facility maintenance domains, with further work expanding on these efforts Akhbardeh et al. (2020a,b). The transformation of text into a KG can be achieved through Information Extraction pipelines, which typically consist of stages such as CR, NEL, RE, and KG construction (Bratanic (2021)). Beyond datasets, the development of shared ontologies and standardized data

formats between different organizations and research groups is crucial to fostering interoperability and accelerating research in technical domains (Meng et al. (2023)). However, it remains difficult for models to paraphrase information when it comes from domains with few resources. (Li et al. (2024)). Therefore, it is import to have tools such as EvalxNLP which highlight how much information can be explained (Dhaini et al. (2025)).

3 Research Problem

Organizations in the aviation and manufacturing domains generate vast amounts of unstructured maintenance records rich with operational insights, yet these records are difficult to analyze due to their inconsistent structure, jargon, and shorthand. KE techniques, such as CR, NER, NEL, and RE, can transform this text into structured formats like knowledge graphs, but most tools are trained on open-domain prose and perform poorly on technical, domain-specific text. Furthermore, prior research highlights the strengths and limitations of current approaches to KE, particularly in the integration of LLMs with KGs. Although LLMs excel in generalizability, they often falter in domain-specific accuracy. This gap presents a major barrier to deploying trusted KE systems in safety-critical settings, where accuracy, reproducibility, and transparency are essential. Progress is further hindered by the lack of public, domain-specific benchmarks; most datasets are proprietary or too narrow for robust evaluation. This work addresses that gap in the aviation and maintenance sectors by introducing the Operations and Maintenance Intelligence benchmark, or OMIn, a benchmark dataset based on FAA incident reports, enabling targeted evaluation of KE tools. In line with open data initiatives (Brundage et al. (2021)), we contribute an open-source dataset and detailed evaluations for zero-shot performance of sixteen KE tools, setting a benchmark for future research and tool development in this domain.

4 Methodology

To evaluate KE approaches in the maintenance domain, we construct a benchmark dataset and systematically assess tool performance across four key tasks: NER, CR, NEL, and RE. Figure 3 illustrates the end-to-end workflow, which begins with the collection and curation of domain-specific data and proceeds through gold standard development and tools selection, and on to implementation and evaluation. Each step is detailed in the subsections that follow or Section 5, Evaluation, as indicated in the diagram.

4.1 Data Selection

While there are few open-source maintenance and operations datasets, we want to highlight four noteworthy sources.

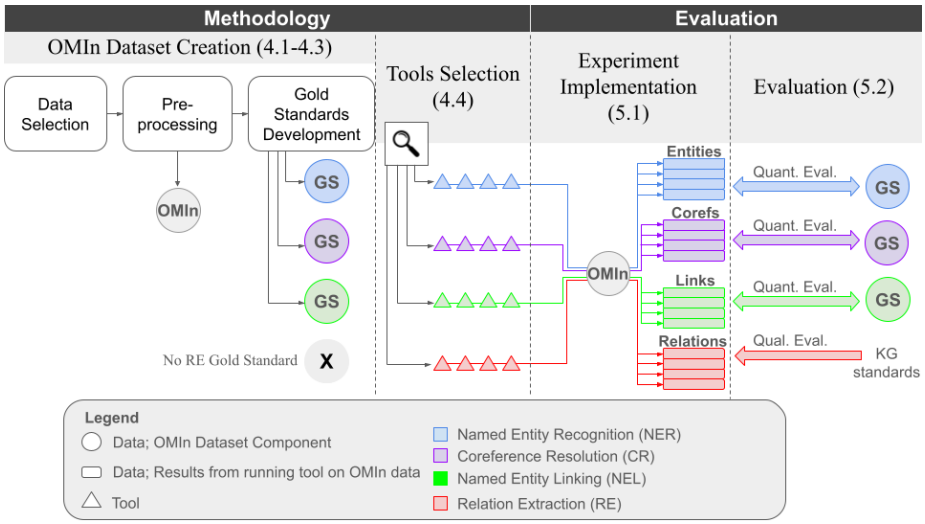

Figure 3: Conceptual Overview of Methodology and Evaluation Strategy Used in This Study. To create the OMIn dataset, we proceed through data selection (Subsection 4.1), pre-processing (Subsection 4.2), and Gold Standards (GSs) development (Subsection 4.3). Then, we select four tools for each stage of the KE workflow (Subsection 4.4) These sixteen resultant tools are then implemented on the OMIn dataset, each creating a set of results in the form of named entities (NER), co-references (CR), linked entities (NEL), or relational triples (RE). The experimental setup for this implementation is captured in Subsection 5.1. These results are then evaluated against their respective GSs, or in the case of RE, against qualitative standards for knowledge representation in KGs. Development and implementation of evaluation metrics is discussed in Subsection 5.2.

- NASA's Prognostics Center of Excellence (PCOE) offers several datasets that track the performance, operating conditions, and indications of damage of components such as bearings and batteries, as well as machines such as millers. However, there is no free-response natural language text in these records (NASA Prognostics Center of Excellence (2023)).
- NASA's Aviation Safety Reporting System (ASRS) maintains a database of aviation safety incident and situation reports. It features a search engine that enables users to filter and download data. Additionally, ASRS provides PDF files containing a small selection of 50 records pertinent to 30 different topics (NASA Aviation Safety Reporting System).
- MaintNet's aviation dataset from the University of North Dakota Aviation Program (2012-2017) has 6,169 records. The records have "Problem" and "Action" fields and describe how problems were fixed. Although the dataset is fairly large, many items are repetitive, brief, and lack narrative context (Akhbardeh et al. (2020a)).

- The Federal Aviation Administration (FAA) maintains a database of aviation accident and incident reports spanning more than 45 years. The Accident & Incident Data (AID) dataset contains reports with a description of each incident along with details such as airplane type, an accident-type code, and more (Federal Aviation Administration (2024)).

After reviewing the available options, we decided to use the FAA's AID dataset for our evaluation. While ASRS offers similar data, it includes numerous redacted proper nouns that could potentially confuse the KE tools. Although MaintNet's "Problem" and "Action" fields could be useful for training a model to aid in problem-solving within the maintenance domain, they lack sufficient narrative and context for effective KE evaluation.

We recognize that AID reports only describe events noticed by the pilot and ground crew during flight operations, not actions taken during direct maintenance. Although we use a subset of the AID dataset consisting of maintenance-related accidents and incidents, the resultant dataset is still distinct from a maintenance dataset made up of logs written by a maintenance technician. However, AID is a valuable starting point for operational and maintenance KE in several key ways: its short document size, frequent use of domain-specific shorthand and acronyms, and use of identification codes for vehicles and system components.

Building on the related work, our study evaluates KE tools using a novel dataset tailored to the maintenance and aviation domains. The following sections introduce this dataset and outline our evaluation methodology, providing insights into the challenges and opportunities for advancing domain-specific KE.

4.2 Dataset Creation: OMIn

We downloaded all available records from the FAA AID dataset spanning from 1975 to 2022 in June 2022, which totaled in excess of 210,000 records.[2] We examined the fields of the records, as detailed in our data documentation, and found that 8 of the 116 incident types were related to maintenance. We selected records belonging to the 8 maintenance-related incident types and excluded those without textual description fields, resulting in a refined dataset of 2,748 records. We did not perform spellchecking or acronym resolution since our initial objective was to establish a baseline using the raw data. We refer to our subset of AID as the Operations and Maintenance Intelligence (OMIn) dataset. We treat each record in the OMIn dataset as a separate document when loading the dataset into the systems we evaluate. Figure 4 shows the distribution of document lengths in OMIn. Figure 5 illustrates how the OMIn dataset is curated from AID.

Some of the tools allowed the OMIn dataset to be passed in directly as plain text. However, other tools were more challenging to run and required the FAA Data to be preprocessed. Table 1 shows which tools allowed data to be passed directly as plain text. The tools trained on CoNLL-2012, which were ASP

[2]The dataset is updated regularly, and at the time of download, the latest record was from May 24, 2022.

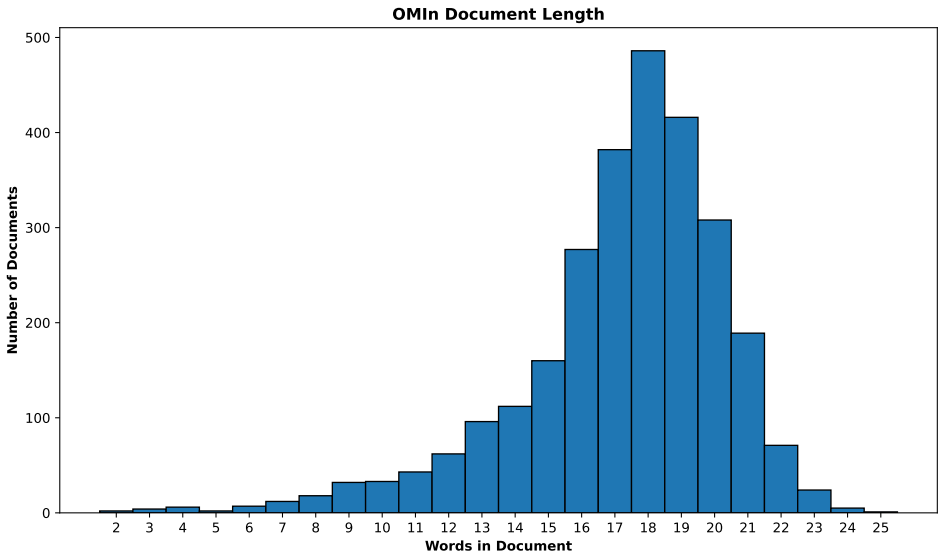

Figure 4: Distribution of Document Lengths in OMIn. The OMIn Dataset features 2748 short documents, usually 1-3 incomplete sentences, which are drawn from accident/incident reports captured in AID. The documents range between 2 and 25 words. The mean is 17.23, and the standard deviation is 3.14. The Q1 is 16; the median is 18; and Q3 is 19.

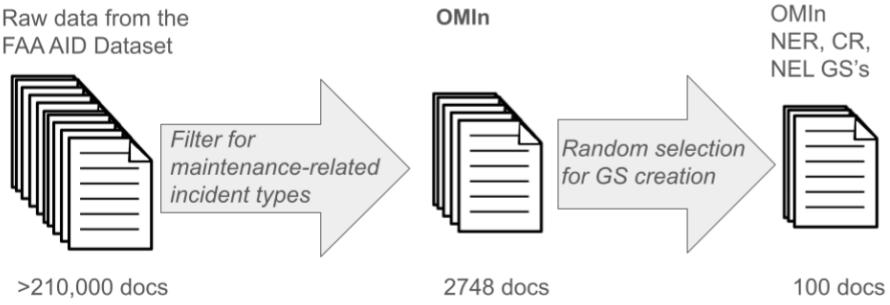

Figure 5: OMIn Dataset Curation. The OMIn Dataset is a subsection of maintenance-related incidents from the FAA Accident/Incident Dataset. A random selection of 100 records from OMIn were chosen as the basis for task-specific gold standards. The remaining 2648 documents in OMIn are un-labeled.

and s2e-coref, required data in an annotated CoNLL-2012 file format rather than simple strings. Therefore, we adapted the FAA data to the CoNLL-2012 format, which organizes documents in a tabular structure with each word on a separate line and more than 11 columns detailing the word's semantic role. See

A for more information on our process.

NER		Coref		NEL		RE	
spaCy	✓	ASP	✗	BLINK	✓	REBEL	✓
flair	✓	coref-mt5	✗	spaCy Entity Linker	✓	UniRel	✓
stanza	✓	s2e-coref	✗	GENRE	✓	DeepStruct	✗
nltk	✓	neuralcoref	✓	ReFinED	✓	PL-Marker	✗

Table 1: Tools That Can Directly Process OMIn Text.

4.3 Gold Standard Development

Gold Standards (GSs) for each KE task can vary depending on the criteria selected by the research team responsible for their creation. Our team began by considering how to structure a maintenance ontology. From there, we tailored the gold standards to align with and augment the maintenance ontology. Due to the significant manual effort this process required, we selected a sample of 100 records from our dataset to create the gold standards. We intend to grow the number of curated records with community collaboration.

Gold Standard vs Ground Truth

We assign the term gold standard versus ground truth as the dataset was human-curated and represents language relations and classifications that are not universally defined nor understood as ground truth measures.

4.3.1 Named Entity Recognition Gold Standard

There are several open-source standards for NER annotation, used in shared tasks such as CoNLL-2003, ACE-2005, and CoNLL-2012. Each of these shared tasks has developed its own set of annotation guidelines and set of named entity types. The entity types used in the CoNLL-2012 NER task correspond to the schema defined in the OntoNotes 5.0 corpus, and the 18 entity types are most commonly referred to as the OntoNotes 5.0 set. We adopt that convention in this paper. Most NER systems classify entities that fall into ENAMEX types such as person, location, and organization, with some also classifying time and date (TIMEX) types and number (NUMEX) types. No NER benchmark focuses on entity types relevant to data from the maintenance or aviation domains. Absent of these types, we gathered a small team of trained annotators who labeled un-typed named entities that constituted the essential information in each record of the FAA dataset. We started by following guidelines for ACE-2005, and added TIMEX and NUMEX entities according to OntoNotes 5.0. We then added additional entities for essential aviation maintenance information, including aircraft parts, systems, phases and types of operations, and equipment failures. We refer to this GS as the Un-Typed FAA (UTFAA) GS and record our annotation guidelines in B.

	CoNLLFAA	ACE05FAA	ACE1FAA	ONFAA	UTFAA
Total	**44**	**195**	**122**	**61**	**509**
PER	3	49	49	3	–
ORG	11	11	11	11	–
LOC	21	15	15	0	–
MISC	9	–	–	–	–
GPE	–	14	14	14	–
FAC	–	33	33	7	–
VEHICLE	–	73	–	–	–
PRODUCT	–	–	–	9	–
QUANTITY	–	–	–	6	–
CARDINAL	–	–	–	3	–
DATE	–	–	–	4	–
TIME	–	–	–	4	–

Table 2: Distribution of Entity Types across NER Gold Standards. The "–" symbol indicates that the Entity Type does not belong to the corresponding standard.

For example, in the record, "Narrative: The cargo door was latched before takeoff by Mr. Bowen. Runway conditions at Steven's Village was extrem," the entities we annotate are "cargo door", "takeoff", "Mr. Bowen", "runway conditions", and "Steven's Village."

We also annotated the FAA dataset following the guidelines used in CoNLL-2003, ACE Phase 1, ACE-2005, and OntoNotes 5.0. We refer to these as the benchmark-annotated GSs when grouped collectively or as CoNLLFAA, ACE1FAA, ACE05FAA, and ONFAA, respectively. ACE1FAA is equivalent to ACE05FAA with the vehicle-type entities removed, which leaves the set of entities which NLTK is trained to recognize. Meanwhile, PL-Marker, whose NER subtask we evaluate, recognizes the set of entities in ACE-2005. A summary of the gold standard entities and their distribution across entity types can be seen in Table 2. Note that the different gold standards may tag entities differently, even for the same entity type. This is due to differences in annotation guidelines across the benchmarks.

We compared the three resulting sets of named entities against our set of named entities and the overlap as recorded in Table 3. Total refers to the total number of entities generated for the set of 100 sample records by each benchmark-annotated GS. Match and Partial Match refer to the number of entities in each benchmark-annotated GS that match or partially match an entity in our GS, regardless of label. Overlap is the sum of the matches and partial matches divided by the total number of entities in our GS, which is 510. For evaluation, we use both our un-typed gold standard and the benchmark-annotated gold standards.

	Total	Match	Partial Match	Overlap
CoNLLFAA	44	36	8	0.086
ACE05FAA	195	133	54	0.35
ACE1FAA	122	89	26	0.22
ONFAA	61	52	9	0.12

Table 3: Agreement of NER on Gold Standard

4.3.2 Coreference Resolution Gold Standard

There were no maintenance-specific adjustments necessary for CR, so we adhered to CoNLL-2012 guidelines and the OntoNotes 5.0 phrase tagging guidelines (Pradhan et al. (2012)). In CoNLL-2012, co-referential entities may include a broad scope of phrases with potentially differing grammatical structures and roles, linked by a common reference to the same real-world entity.

4.3.3 Named Entity Linking Gold Standard

Our NEL gold standard is based on the named entities identified in our UTFAA gold standard. We found Wikidata Q-identifiers (QIDs) by manually looking up each entity and listing the most specific Q-identifier if there was a correct one. All of the NEL tools we evaluate use QIDs or Wikipedia-based identifiers, such as titles or links to entries, which can be easily translated into QIDs. This enabled us to directly compare the links predicted by each NEL tool with those in the gold standard.

We also created a Flexible NEL GS, which includes additional entity-QID links, motivated by the fact that differing mention spans may change the appropriate QID for each entity. For example, in the sentence "While taxiing lost nosewheel steering and brakes", we have "nosewheel steering" as an entity in our UTFAA NER GS. If an NEL tool only recognizes "steering" as the entity and links it to the QID for steering correctly (Q18891017), this would be excluded from the evaluation set under strong entity-matching and counted as incorrect under weak entity-matching. Our Flexible GS makes a flexible evaluation possible, where if an exact match for "nosewheel steering" is not found, the evaluator moves to a secondary entity-QID link, ("steering", Q18891017), and evaluates the predicted entity-QID against it. To accomplish this, we included primary, secondary, and up to tertiary entity-QID pairs for entities such as "nosewheel steering," where sub-spans of the primary entity share the same semantic role in the sentence.

4.3.4 Absence of Relation Extraction Gold Standard

Unlike the other KE tasks considered in this study, we do not provide a gold standard (GS) for RE. The primary impediment to establishing such a GS is the heterogeneity of the relations that different tools recognize. Each evaluated tool is trained on a distinct set of relationships which Table 4 highlight showing illustrative training data relationships for each RE model, as well as a subset

Training Data	Tool	# Relations	Sample Relations
Wikidata	REBEL	220	has part, part of, has effect, has cause, location, subclass of, ...
NYT	UniRel, DeepStruct	25	location/contains, person/company, company/founders, ...
ACE05	PL-Marker	6	PER-SOC, ART, PHYS, ORG-AFF, ...
SciERC	PL-Marker	7	PART-OF, USED-FOR, FEATURE-OF, CONJUNCTION, ...

Table 4: RE Tool Models Predict Different Relations Depending on Training Data

of the relation sets they employ. Although some conceptual overlap exists, no unified ontology currently aligns these disparate sets into a coherent framework that could serve as the basis for a universal gold standard.

In principle, one could construct a specialized maintenance ontology that enumerates domain-relevant relations and subsequently benchmark each tool against it. However, such an approach introduces several methodological complications. Relations often differ substantially in their level of granularity and intended scope across datasets. For example, ACE-2005 employs a high-level "PART-WHOLE" relation that encompasses both geographic and non-geographic entities, while the New York Times (NYT) corpus relies on a narrower "location contains" relationship. In contrast, Wikidata's ontology includes multiple properties—such as "located in the administrative territorial entity," "part of," and "location"—that articulate spatial and hierarchical relations at different levels of specificity. Selecting any one level of granularity for a putative gold standard would inevitably privilege certain tools' relational inventories and disadvantage others. Moreover, introducing relations absent from a given tool's training data would yield uniformly poor performance on those relations, thereby obscuring meaningful comparisons across tools.

The only rigorous way to ensure fairness and consistency would be to fine-tune all RE models on a common, domain-specific ontology. Yet, constructing such an ontology and annotating a sufficiently large corpus of maintenance text represents a significant investment of time and effort. Furthermore, expert input from the maintenance community would be essential for achieving conceptual clarity and operational relevance. We thus consider this endeavor more suitably reserved for future collaborative work rather than as part of the present study.

Consequently, we do not report an F1 score against a unified GS for RE. Instead, as described in Section 5.2.4, we rely on accuracy and other metrics that do not presuppose a single ontology. We anticipate that subsequent studies, supported by broader community engagement, will produce a maintenance-specific ontology and corresponding gold standard that enable robust, equitable evaluations of RE performance.

4.4 Tools Selection

We selected sixteen tools for evaluation, with four dedicated to each KE task. The primary criterion guiding this selection was the availability of off-the-shelf, open-source solutions specifically fine-tuned and benchmarked for NER, CR, NEL, and RE. This approach enables us to compare the tools' expected in-

Tool	Entity Set	Benchmark Performance (F1)
spaCy EntityRecognizer	OntoNotes 5.0	Benchmark Not found
Flair NER	CoNLL-2003, Ontonotes 5.0	94.09 (CoNLL-03), 90.93 (OntoNotes)
Stanza NERProcessor	CoNLL-2003, Ontonotes 5.0	92.1 (CoNLL-03), 88.8 (OntoNotes)
NLTK ne_chunk	ACE Phase 1 + GSP[3]	No Benchmark Found

Table 5: NER Tools Selected for Evaluation

domain performance with their observed zero-shot results on our maintenance dataset, thereby providing a stable and interpretable baseline.

Although it is true that the latest LLMs, including transformer-based systems such as GPT variants or LLaMA, have demonstrated state-of-the-art performance in various NLP tasks, their general-purpose capabilities often come at the cost of increased complexity in domain-specific applications. Adapting these models for KE tasks currently involves additional steps, such as prompt engineering, specialized prompting frameworks, and robust guardrails to mitigate unpredictable outputs. Although these LLMs excel at open-ended reasoning and question-answering, they lack the immediate, task-specific fine-tuning that would allow for straightforward evaluation in our zero-shot setting.

Our decision to focus on models already proven in KE benchmarks, many of which have transparent code bases and straightforward deployment procedures, aligns with our emphasis on trusted and reproducible AI solutions. It also establishes a clear starting point. By first understanding how specialized KE tools perform without adaptation, we create a benchmark against which future efforts, such as applying domain adaptation, fine-tuning, or integrating cutting-edge LLMs, can be measured. Thus, our findings lay the groundwork for subsequent experimentation with advanced transformer-based models, once their methods for reliably performing KE tasks are more thoroughly developed.

Additionally, we excluded models exceeding 13B parameters, as such large architectures typically require substantial computational resources, often facilitated through cloud-based services. Although state-of-the-art in many tasks, these models introduce practical concerns related to infrastructure, cost, and security, particularly for organizations handling sensitive and confidential data. By highlighting tools that can be run on-premises and without extensive cloud dependencies, we underscore our commitment to delivering trustworthy, secure solutions aligned with operational needs in the maintenance domain.

The website paperswithcode.com was a helpful resource for finding open-source models that have reported high scores on popular benchmarks (Meta).

4.4.1 Named Entity Recognition Tools

Since there are a variety of high-performing NER tools available, we prioritized evaluating readily available tools over those with the highest performance. This led us to the well-known NLP toolkits spaCy (Honnibal et al. (2020)), flair (Akbik et al. (2018)), stanza (Shan et al. (2023)), and NLTK. Table 5 summarizes the NER tools selected.

4.4.2 Coreference Resolution Tools

Many coreference resolution tools have been bench-marked on well-recognized shared tasks and challenges such as CoNLL-2012 (Pradhan et al. (2012)), Gendered Ambiguous Pronouns (GAP) (Webster et al. (2019)), and the Winograd Schema Challenge (WSC) (Levesque et al. (2012)). GAP and WSC evaluate a model's ability to link pronouns to their antecedents. However, CoNLL-2012 includes noun phrases that refer to one another, such as "vehicles" and "armored vehicles." Since maintenance data rarely includes pronouns and often refers to system components in multiple ways, we decided that tools trained for CoNLL-2012 would be the best fit to evaluate maintenance data. The tools selected for this evaluation are summarized below.

- *Autoregressive Structured Prediction with Language Models (ASP)* is a T5-based model developed by Google Research in 2022 (Liu et al.). ASP proposes a conditional language model trained over structure-building actions instead of strings. This enables it to capture the structure of the sentence more effectively and build the target structure step by step. It performs NER, CR, and RE. Its CR models include three based on different sizes of flant5 (base, large, and xl), and one based on T0-3B. It achieved an F1 score of 82.3 for CR n on CoNLL-2012.

- *Coref_mt5* is a mT5 model developed by Google Research in the work *"Coreference Resolution through a seq2seq Transition-Based System"* (Bohnet et al. (2022)). This model leverages the mT5 architecture and employs a seq2seq approach. It encodes a single sentence along with its preceding context as input and generates an output with predicted coreference links. The model utilizes a transition-based system to extend discovered coreference chains and establish new ones in subsequent sentences. It achieved an F1 score of 83.3 on CoNLL-2012.

- *Start-To-End Coreference Resolution (s2e-coref)* introduces a lightweight approach that avoids constructing span representations Kirstain et al. (2021). Instead, it uses contextualized representations of the boundaries of spans to score the likelihood of a coreference between a mention and potential antecedents. It is based on longformer-large and achieved an F1 score of 80.3 on CoNLL-2012.

- *NeuralCoref* is a coreference resolution pipeline component developed by the spaCy team. It uses the spaCy parser for mention-detection and ranks possible mention-coreference pairs using a feedforward neural network developed by Clark and Manning, Stanford University. The Clark and Manning network achieved an F1 score of 74.23 on CoNLL-2012 in 2016.

[3]NLTK ne_chunk recognizes the five entity types found in ACE Phase 1 (Consortium (2002)), as well as a sixth type, Geographical-Social-Political Entity (GSP), an early form of Geo-Political Entity (GPE), found in ACE-Pilot (Consortium (2000)).

4.4.3 Named Entity Linking Tools

We chose three NEL tools that have achieved state-of-the-art results and are compatible with Wikidata. Additionally, we decided to evaluate the spaCy EntityLinker because of spaCy's wide recognition.

- *Better Entity LINKing (BLINK)* introduces a two-stage zero-shot entity linking algorithm, with a bi-encoder for dense entity retrieval and a cross-encoder for re-ranking (Wu et al. (2020)). It uses a predefined catalog of entities from Wikipedia and uses the first few sentences of their Wikipedia summaries as context. It uses BERT as a base model for its encoders. BLINK does not do NER on its own but utilizes flair. It achieved a 76.58% accuracy on the Zero-shot EL dataset and a 94.5% accuracy on TACKBP-2010.

- *spaCy EntityLinker* is spaCy's NEL pipeline component. It uses InMemoryLookupKB to match mentions with external entities. InMemoryLookupKB contains Candidate components that store basic information about their entities, like frequency in text and possible aliases.

- *Generative ENtity REtrieval (GENRE)* employs a BART-based seq2seq model to autoregressively generate entity identifiers (De Cao et al.). Additionally, GENRE uses a constrained decoding strategy that forces each generated identifier to be in a predefined candidate set, ensuring that the generated output is a valid entity name. It achieved micro-F1 scores ranging from 77.3 to 94.3 on both in-domain and out-of-domain benchmarks.

- *Representation and Fine-grained typing for Entity Disambiguation (ReFinED)* uses fine-grained entity types and entity descriptions to perform mention detection, fine-grained entity typing, and entity disambiguation in a single forward pass (Ayoola et al.). Similar to BLINK, ReFinED uses a catalog of entities from Wikipedia (and Wikidata) with context from their summaries, and new entities can be added to the catalog without retraining. ReFinED includes three NEL models, wikipedia_model, wikipedia_model_with_numbers, and aida_model, all based on RoBERTa. It achieves micro-F1 scores ranging from 78.2 to 94.8 on both in-domain and out-of-domain benchmarks.

4.4.4 Relation Extraction Tools

We chose the following Relationship Extraction (RE) tools because they had reached state-of-the-art performance on widely recognized benchmarks, including CoNLL-2004, NYT, and ACE-2005. Note that each benchmark dataset uses a different set of relations, depending on the subject of the data. None of these sets of relations were adequate to capture all of the most relevant information for maintenance data. Note that reference to relational triples or triplets below refer to structures in the form (*subject*, *relation*, *object*).

- *Relation Extraction By End-to-end Language generation (REBEL)* uses an autoregressive seq2seq model based on BART to express relation triplets as a sequence of text (Huguet Cabot and Navigli (2021)). It finds 220 relation types, a subset of Wikidata properties chosen by the REBEL team. REBEL achieves an F1 of 91.76 on NYT, 71.97 on CoNLL-2004, and 90.39 on Re-TACRED.

- *Unified Representation and Interaction for Joint Relational Triple Extraction (UniRel)* jointly encodes entities and relations and captures interdependencies between entity-entity interactions and entity-relation interactions through the proposed Interaction Map (Tang et al.). It is based on bert-base-cased, and trained on the New York Times (NYT) dataset. It consists of 24 relation types. UniRel achieves an F1 of 93.7 on NYT and 94.7 on WebNLG.

- *DeepStruct* (Wang et al.) introduced a model pretrained on task-agnostic corpora to enhance language models' ability to generate structured outputs, such as entity-relation triples, from the text. DeepStruct is a unified model designed to generalize across a wide range of datasets and tasks, including RE, NER, and event extraction, among others. However, some steps need to be followed to perform inference on a dataset that is not included in DeepStruct's list of pretrained datasets. The dataset schemes related to the RE task used in the pretrained model are from the datasets NYT, CoNLL04, ADE, and ACE2005. We selected the NYT setup to use in our evaluation, with the highest RE F1 score of 84.6. Additionally, both UniRel and DeepStruct use the NYT dataset, which improves consistency between tools.

- *Packed Levitated Marker (PL-Marker)* uses markers in the encoding phase to capture the interrelation between span pairs (Ye et al. (2022)). The novelty of PL-Marker's approach is its neighborhood-oriented span-packing strategy. PL-marker provides NER and RE models trained on the SciERC and ACE-2005 datasets, as well as three other NER models. Their BERT-based models, which we evaluate, achieved a 69.0 F1 score on ACE-2005 and a 53.2 F1 score on SciERC in RE. We also evaluate their Albert-xxl-based model which achieved state-of-the-art on ACE-2005 RE with an F1 score of 73.0.

5 Evaluation

Our evaluation process focuses on the tools' ability to handle unseen elements in a zero-shot scenario. Zero-shot here refers to evaluating tools on the FAA dataset without prior fine-tuning or specific training in aviation or maintenance domains.

5.1 Experimental Setup

We followed the guidelines provided in each tool's documentation to set up and generate output. To reproduce our results, see our step-by-step methods documented in our repository (Mealey et al. (2024)).

While some tools were straightforward to implement, others presented challenges. The rapid pace of advancements in NLP means that library dependencies quickly become outdated. There can also be a complicated environment resolution setup if the accompanying requirements files do not specify all necessary package versions. Other challenges included a lack of clear indications of the GPU requirements needed for model deployment, and occasional bugs. The greatest challenge was the lack of clear documentation on how to use these tools with our data. This required us to carefully review the tool code base to understand the required data format and identify any necessary adjustments for our dataset compared to the benchmark data. In our GitHub documentation for each tool, we offer a "Reproducibility Rating" along with an account of the challenges we encountered during setup.

For tools that ship multiple models, we strove to evaluate the models most applicable to our task. For spaCy tools, we selected the small and large models. For RE tools, we select the best-performing model as well as others we believe would be informative to evaluate. All models are trained on English data. Table 6 summarizes the models implemented for each tool that has more than one model available.

Tool	**Models Implemented**	**Models Not Implemented**
flair	CoNLL-03, OntoNotes	-
spaCy EntityRec	en_core_web_sm, en_core_web_lg	en_core_web_md, en_core_web_trf
stanza	conll03_charlm, all ontonotes and ontonotes-ww-multi models	conll03_electra-lg, all non-conll03 & ontonotes
ASP	flant5_base, flant5_large, flant5_xl, t0_3b	all non-CR models
neuralcoref	en_core_web_sm, en_core_web_lg	en_core_web_md, en_core_web_trf
BLINK	biencoder, crossencoder	-
spaCy EntityLink	en_core_web_sm, en_core_web_lg	en_core_web_md, en_core_web_trf
GENRE	BART_base E2E EL	-
ReFinED	wikipedia_model_with_numbers, wikipedia_model, aida_model	-
UniRel	NYT	WebNLG
PL-Marker	scibert-uncased (SciERC), bert (ACE-05), albert-xxl (ACE-05)	NER models w/o RE

Table 6: Models Implemented For Each Tool

REBEL provided one English-only model, rebel-large. However, REBEL allows this model to be used in several ways. Rebel-large can be downloaded directly and implemented with user-visible hyper-parameters, or loaded as a component in either a transformers or spaCy pipeline. We chose to test the direct implementation of the model as well as the transformers pipeline. In our testing, the transformers pipeline produced consistent results, while the direct implementation did not, even with the same hyper-parameters.

For the Coref_mt5, a T5X checkpoint from the top-performing mT5 model (Bohnet et al. (2022)) was made publicly available[4]. However, we encountered issues when attempting to load this checkpoint. To address this, we utilized a PyTorch-converted version of the model, which was released by Ian Porada on HuggingFace.[5]

DeepStruct required some additional steps to prepare it for inference on unseen datasets. First, the unseen dataset's schema must be aligned with the one DeepStruct was trained on to ensure compatibility with the model's recognized entity and relation types. Additionally, configuration files might need adjustments to properly format the input data. Considering the RE task, DeepStruct schemas are based on datasets like NYT, CoNLL-04, ADE, and ACE-2005. We chose the NYT schema for our evaluation, which produced the highest RE F1 score of 84.6. This also ensures consistency and fairness when comparing DeepStruct with UniRel.

5.2 Evaluation Metrics

We quantitatively evaluated NER, CR, and NEL tools by comparing their outputs to our established gold standard. In contrast, due to the lack of a gold standard for RE tools in the FAA dataset, we employed manual evaluation by a domain expert. This evaluation was guided by the qualitative criteria outlined by Hogan et al. (2021), with a focus on syntactic accuracy, semantic accuracy, and consistency. Further details on the evaluation methodology are provided in Section D. Additionally, we report the total number of triples generated across the entire FAA dataset, as well as the percentage of records for which triples were generated.

5.2.1 Named Entity Recognition Evaluation

We evaluated each NER tool against the un-typed UTFAA as well as the benchmark-annotated GS corresponding to its training data (e.g., CoNLLFAA for models trained on CoNLL-2003).

For UTFAA, we evaluated entity spans and ignored labels. We report an F1 score with strong-matching and one with weak-matching. In strong-matching, a predicted entity is correct only if it exactly matches a gold standard entity. In weak-matching, a predicted entity is counted correctly if it contains any substring of the gold standard entity, or if the gold standard entity contains any substring of the predicted entity.

In the example: "Narrative: The cargo door was latched before takeoff by Mr. Bowen. Runway conditions at Steven's Village was extreme," the entities should be "cargo door", "takeoff", "Mr. Bowen", "runway conditions", and "Steven's Village". However, flair returns "Bowen" and "Steven's Village" as the entities. Therefore, "Bowen" is marked as incorrect in the strong-matching

[4]https://github.com/google-research/google-research/tree/master/coref_mt5
[5]https://huggingface.co/ianporada/link-append-xxl

evaluation, but correct in the weak-matching evaluation since it is a substring of "Mr. Bowen."

For the benchmark-annotated GSs, we follow SemEval (Segura-Bedmar et al. (2013)) in reporting four F1 metrics: Strict, Type, Exact, and Partial. In Strict and Type, an entity must be labeled with the correct type to be correct. In Strict and Exact, an entity must exactly (strong) match the gold standard entity to be correct, while in Type and Partial, an entity may be a partial (weak) match. Note that Exact and Partial are equivalent to the label-agnostic strong and weak matching used for UTFAA, respectively.

We utilize work by Batista (2018) to implement both NER evaluations.

5.2.2 Coreference Resolution Evaluation

For CR, we report 4 metrics. This is useful for data sets that are limited in size and is appropriate since maintenance data sets have few coreferences. Our FAA CR Gold Standard only has 18 coreferences, so it is important to have a variety of evaluation metrics.

- MUC (Vilain et al. (1995)) measures measures link-based correctness.
- B-CUBED (Bagga and Baldwin (1998)) evaluates the accuracy of individual elements in the clusters.
- LEA (Moosavi and Strube (2016)) considers the similarity between the predicted and the true clusters which considers precision and recall on a cluster level.
- CEAF (Luo (2005)) focuses on the links and entities, considering the impact of each decision within the cluster.

We also follow CoNLL-2012 in reporting an unweighted average of MUC, B-CUBED, and CEAF (Pradhan et al. (2012)), and we employed a coreference evaluation tool, *corefeval*, to conduct our analysis[6].

5.2.3 Named Entity Linking Evaluation

We evaluate NEL tools in two ways: F1 score and Ontology-based Topological (OT) metrics that use features derived from the structure (topology) of the underlying base ontology.

To calculate the F1 score, we follow Shen et al. (2021) and Usbeck et al. (2015). In addition to the F1 score, we used two OT similarity computations over Wikidata available in the KGTK Semantic Similarity system (USC-ISI-I2 (2021)): Jiang Conrath (JC) and class similarity. More details on how the F1 and OT metrics scores are computed are available in C.

Examples of JC and class similarity computations are presented in Table 7, which shows the similarity values between the concept c_1 = *aviation fuel* (Q1875633) and other c_2 concepts, such as *aviation fuel* (Q1875633), *combustible*

[6]corefeval: https://github.com/tollefj/coreference-eval

matter (Q42501), *Fuel* (Q15766923), and *Fuel* (Q5507117). Note that although some concepts include the word *Fuel* in their label, they receive the lowest scores due to their semantic distance from the c_1 concept, *aviation fuel*, as indicated by their values under the column description.

QID	Description	Label	Class	JC
Q1875633	propellents used to power aircraft or aviation...	aviation fuel	1.00	1.00
Q42501	any material that stores energy that can...	combustible matter	0.68	0.89
Q15766923	scientific journal	Fuel	0.03	0.06
Q5507117	short-lived Bay Area post-hardcore musical...	Fuel	0.00	0.00

Table 7: Example OT Measurements

For each metric, we also implement three evaluation strategies: strong-matching, weak-matching, and flexible. Strong and weak matching follow the definitions described in Section 5.2.1. The type of matching affects which predicted entities are included in the evaluation set. Flexible evaluation evaluates predicted entities that have a strong match with entities in the Flexible NEL GS, as described in Section 4.3.3. More details on Flexible evaluation are included in C.

5.2.4 Relation Extraction Evaluation

We evaluate the RE output qualitatively on the same set of 100 records used to create gold standards for the other KE tasks. The relations extracted by an RE system may take several structures, including relational tables, XML files, and relational triples. For our evaluation, we use relational triples, since they are compatible with KG construction technologies such as Neo4j. We evaluate the triples generated for each document as if they were components of a KG, and thus use metrics defined for KGs. We report syntactic accuracy, semantic accuracy, and consistency based on definitions in Chapter 7 of Knowledge Graphs (Hogan et al. (2021)), and further described in D. We also report the total number of triples generated for the entire FAA dataset and the percent of records with generated triples.

6 Results

6.1 Overview

As described in Section 5, we have quantitative evaluations for NER, CR, and NEL as well as qualitative scores for RE. Table 8 shows the main results.[7] More detailed results follow in the respective subsections, including precision, recall, and task-specific metrics.

[7]All items in parentheses following the name of the tool are model names unless otherwise stated. For ReFinED, wiki_w_nums is an abbreviation for the model, wikipedia_model_with_numbers, and wikipedia refers to the entity set used.

NER - UTFAA Strong F1		CR - CoNLL-12 F1	
NLTK ne_chunk	**0.27**	**s2e-coref**	**0.8**
flair (OntoNotes)	0.22	**ASP (flant5-large / t0-3b)**	**0.8**
spaCy EntityRecognizer (en_core_web_lg)	0.17	neuralcoref (en_core_web_lg)	0.5
stanza (ontonotes_electra-large)	0.16	coref-mt5	0.3
NEL - Strong F1		RE - Combined Acc	
spaCy EntityLinker (lg)	**0.20**	**PL-Marker (albert-xxl)**	**1.0**
BLINK (biencoder)	0.14	DeepStruct (NYT)	0.7
ReFinED (wiki_w_nums, wikipedia)	0.10	REBEL	0.58
GENRE	0.0063	UniRel (NYT)	0.083

Table 8: NLP Tool Zero-Shot Scores on FAA Data

6.2 Named Entity Recognition

As described in Section 5.2.1, we have strong-match and weak-match F1 scores for the NER tools which can be seen in the tables below. Table 9 shows the label-agnostic results from evaluation on UTFAA, and Tables 10, 11, 12, and 13 show the SemEval F1 scores on the benchmark-annotated datasets. Each tool is evaluated on the benchmark-annotated dataset that corresponds to the set of entities it is trained to recognize, which is denoted in brackets. Note that we also provide scores for PL-Marker, since it performs NER as an intermediate step in RE, and outputs its named entities in a readily available file.

NLTK ne_chunk is very sensitive to case. It can find entities well if given input with sentence-casing, including capitalized proper nouns. However, the FAA data is upper-cased by default, which loses the distinction between words that are naturally upper-cased or capitalized with those that are not. We input the data to ne_chunk in lowercase and uppercase. It only recognized entities if the text was inputted in uppercase, and provided the label "ORGANIZATION" for every one. This reduces trust in its overall effectiveness.

6.3 Coreference Resolution

As described in Section 5.2.2, we have four different F1 scores for CR. Additionally, we have the CoNLL-2012 F1, which is the unweighted average of the F1 scores from MUC, B-CUBED, and CEAF. See Table 14 for results. ASP flant5-large and t0-3b happen to have the same output on our evaluation sample; their repeated scores are not a mistake. We only report one significant figure because there are only eighteen records with any coreferences in our gold standard.

6.4 Named Entity Linking

As described in Section 5.2.3, we have three different evaluations for NEL, strong-matching, weak-matching, and flexible. We show the results for each evaluation below.[8] Tables 15 compares tools' performance on an F1 metric,

[8]For ReFinED, wiki_w_nums is an abbreviation for the model, wikipedia_model_with_numbers, wiki is an abbreviation for the model wikipedia_model,

	Weak			Strong		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**
PL-Marker (ACE-2005 bert)	0.78	0.27	**0.4**	0.68	0.24	**0.35**
NLTK ne_chunk (uppercased)	0.43	0.37	**0.4**	0.29	0.25	0.27
spaCy EntityRecognizer (en_core_web_lg)	0.6	0.16	0.26	0.39	0.11	0.17
flair (OntoNotes)	0.68	0.16	0.25	0.59	0.14	0.22
PL-Marker (ACE-2005 albert-xxl)	0.79	0.14	0.24	0.7	0.12	0.21
spaCy EntityRecognizer (en_core_web_sm)	0.56	0.13	0.21	0.36	0.084	0.14
stanza (ontonotes_electra-large)	0.73	0.1	0.18	0.66	0.094	0.16
stanza (ontonotes-ww-multi_electra-large)	0.59	0.096	0.17	0.51	0.082	0.14
stanza (ontonotes_nocharlm)	0.63	0.075	0.13	0.44	0.053	0.094
PL-Marker (SciERC scibert-uncased)	0.69	0.074	0.13	0.46	0.049	0.089
flair (CoNLL-2003)	0.77	0.064	0.12	0.64	0.053	0.098
stanza (ontonotes-ww-multi_nocharlm)	0.49	0.071	0.12	0.38	0.055	0.096
stanza (ontonotes_charlm)	0.71	0.066	0.12	0.51	0.047	0.086
stanza (ontonotes-ww-multi_charlm)	0.64	0.045	0.084	0.42	0.029	0.055
stanza (conll03_charlm)	0.54	0.04	0.075	0.42	0.031	0.059
NLTK ne_chunk (lowercased)	0.0	0.0	–	0.0	0.0	–

Table 9: NER UTFAA Evaluation Results

	STRICT			EXACT			PARTIAL			TYPE		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
flair (CoNLL-03)	0.43	0.41	**0.42**	0.45	0.43	**0.44**	0.54	0.51	**0.52**	0.52	0.50	**0.51**
stanza (conll03_charlm)	0.21	0.18	0.20	0.24	0.20	0.22	0.33	0.28	0.30	0.34	0.30	0.32

Table 10: NER CoNLLFAA Evaluation Results

	STRICT			EXACT			PARTIAL			TYPE		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
stanza (ontonotes_electra-large)	0.59	0.70	**0.64**	0.60	0.72	**0.66**	0.65	0.78	**0.71**	0.66	0.79	**0.72**
stanza (ontonotes-ww-multi_electra-large)	0.41	0.55	0.47	0.42	0.56	0.48	0.49	0.66	0.57	0.51	0.68	0.58
flair (OntoNotes)	0.32	0.61	0.42	0.36	0.68	0.47	0.40	0.77	0.53	0.40	0.76	0.52
stanza (ontonotes_charlm)	0.32	0.25	0.28	0.34	0.26	0.30	0.44	0.34	0.38	0.43	0.33	0.37
stanza (ontonotes_nocharlm)	0.23	0.22	0.23	0.31	0.30	0.31	0.41	0.40	0.40	0.30	0.29	0.29
stanza (ontonotes-ww-multi_nocharlm)	0.22	0.25	0.23	0.26	0.30	0.28	0.36	0.42	0.39	0.28	0.33	0.31
stanza (ontonotes-ww-multi_charlm)	0.31	0.18	0.22	0.36	0.21	0.27	0.57	0.33	0.42	0.56	0.32	0.41
spaCy EntityRecognizer (en_core_web_sm)	0.10	0.19	0.13	0.18	0.35	0.24	0.24	0.46	0.31	0.16	0.31	0.21
spaCy EntityRecognizer (en_core_web_lg)	0.071	0.16	0.098	0.14	0.32	0.20	0.18	0.41	0.25	0.092	0.21	0.13

Table 11: NER ONFAA Evaluation Results

	STRICT			EXACT			PARTIAL			TYPE		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
NLTK ne_chunk (uppercased)	0.0066	0.022	0.01	0.087	0.30	0.13	0.13	0.44	0.20	0.024	0.081	0.037
NLTK ne_chunk (lowercased)	0.00	0.00	–	0.00	0.00	–	0.00	0.00	–	0.00	0.00	–

Table 12: NER ACE1FAA Evaluation Results

	STRICT			EXACT			PARTIAL			TYPE		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
PL-Marker (ACE-2005 bert)	0.53	0.47	**0.50**	0.54	0.49	**0.51**	0.67	0.60	**0.64**	0.77	0.69	**0.73**
PL-Marker (ACE-2005 albert-xxl)	0.62	0.28	0.39	0.62	0.28	0.39	0.78	0.35	0.49	0.92	0.42	0.58

Table 13: NER ACE05FAA Evaluation Results

and aida is an abbreviation for the model aida_model. Wikipedia and Wikidata refer to the entity sets used in inference.

	MUC			B-cubed			CEAF			CoNLL-12	LEA		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**F1**	**rec**	**Rec**	**F1**
s2e-coref	0.9	0.7	**0.8**	0.9	0.7	**0.8**	0.9	0.7	**0.8**	**0.8**	0.9	0.7	**0.8**
ASP (t0-3b)	0.7	0.7	0.7	0.7	0.8	**0.8**	0.8	0.8	**0.8**	**0.8**	0.7	0.7	0.7
ASP (flant5-large)	0.7	0.7	0.7	0.7	0.8	**0.8**	0.8	0.8	**0.8**	**0.8**	0.7	0.7	0.7
ASP (flant5-xl)	0.7	0.6	0.7	0.8	0.6	0.7	0.8	0.6	0.7	0.7	0.7	0.6	0.6
ASP (flant5-base)	0.7	0.5	0.6	0.7	0.5	0.6	0.7	0.6	0.6	0.6	0.7	0.5	0.6
neuralcoref (en_core_web_lg)	0.6	0.4	0.5	0.6	0.4	0.5	0.6	0.5	0.5	0.5	0.6	0.4	0.5
coref_mt5	0.8	0.2	0.3	0.9	0.2	0.3	0.8	0.2	0.3	0.3	0.8	0.2	0.3
neuralcoref (en_core_web_sm)	0.1	0.1	0.1	0.2	0.1	0.1	0.3	0.2	0.2	0.1	0.1	0.1	0.1

Table 14: Coreference Resolution Quantitative Evaluation Results

and Table 16 compares tools' performance on the OT metrics, JC and Class similarity. Note that the F1 metric results in a very different ranking than the OT metrics. This is due to the influence of recall on F1. spaCy EntityLinker has a high recall on the entities in our GS, but does not correctly link them at as high a rate than most of the other tools. Since the OT metrics only compare valid predicted entities against the gold standard, missing gold standard entities have no bearing on the score.

Note that the results for ReFinED's aida_model are close to zero for F1 scores since it only recognizes and links eight entities in the sample records. The OT scores, then, should be understood as circumstantial to the eight entities found, and not a reliable indication of the aida_model's performance.

	Weak			Strong			Flex		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
spaCy EntityLinker (en_core_web_lg)	0.19	0.20	**0.19**	0.15	0.30	**0.20**	0.15	0.18	**0.16**
spaCy EntityLinker (en_core_web_sm)	0.17	0.21	**0.19**	0.14	0.30	**0.19**	0.15	0.17	**0.16**
BLINK (biencoder)	0.64	0.068	0.12	0.57	0.08	0.14	0.64	0.052	0.096
BLINK (crossencoder)	0.61	0.065	0.12	0.52	0.074	0.13	0.61	0.05	0.092
ReFinED (wiki_w_nums, wikipedia)	0.67	0.058	0.11	0.53	0.056	0.10	0.71	0.049	0.092
ReFinED (wiki_w_nums, wikidata)	0.72	0.058	0.11	0.57	0.056	0.10	0.76	0.049	0.092
ReFinED (wiki, wikidata)	0.35	0.03	0.055	0.30	0.041	0.073	0.36	0.028	0.051
ReFinED (wiki, wikipedia)	0.33	0.03	0.055	0.29	0.041	0.073	0.35	0.028	0.051
GENRE	0.12	0.0032	0.0063	0.059	0.0033	0.0063	0.20	0.0045	0.0087
ReFinED (aida, wikidata)	0.0	0.0	0.0	0.17	0.0032	0.0063	0.0	0.0	0.0
ReFinED (aida, wikipedia)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 15: NEL F1 Quantitative Evaluation Scores

6.5 Relation Extraction

As described in Section 5.2.4, and D, we qualitatively evaluated RE for syntactic accuracy, semantic accuracy, and consistency. These scores can be found in Table 17, which orders the tools by number of triples evaluated, and highlights high-scorers in bold. Note that the rightmost column "% Docs w/ Predicted Trip" denotes the percentage of the 2748 OMIn dataset records for which the corresponding tool extracted one or more triples.

Since REBEL extracts at least ten times more triples than the other tools, its syntactic, semantic, and consistency numbers are more meaningful. This in part due to REBEL's set of 220 relations, which is far larger than the sets of 6-25 relations on which the other RE tools are trained. However, the 6

	Weak		Strong		Flex	
	JC	Class	JC	Class	JC	Class
ReFinED (wiki_w_nums, wikipedia)	**0.89**	**0.87**	**0.84**	**0.81**	**0.90**	**0.88**
ReFinED (wiki_w_nums, wikidata)	**0.89**	**0.87**	0.80	0.78	**0.90**	**0.89**
ReFinED (wiki, wikidata)	0.84	0.74	0.74	0.65	0.82	0.72
ReFinED (aida, wikidata)	0.84	0.48	0.73	0.5	0.61	0.33
BLINK (biencoder)	0.82	0.77	0.68	0.64	0.8	0.75
BLINK (crossencoder)	0.82	0.77	0.64	0.60	0.8	0.75
ReFinED (wiki, wikipedia)	0.80	0.71	0.71	0.62	0.78	0.69
ReFinED (aida, wikipedia)	0.56	0.32	0.49	0.25	0.46	0.25
GENRE	0.32	0.28	0.27	0.20	0.37	0.33
spaCy EntityLinker (en_core_web_sm)	0.15	0.073	0.14	0.071	0.15	0.082
spaCy EntityLinker (en_core_web_lg)	0.15	0.082	0.13	0.069	0.14	0.077

Table 16: NEL OT Quantitative Evaluation Scores

most common relations make up 60% of the generated triples and the 25 most common relations make up 92% of the generated triples, so the greater number of possible relations does not fully account for the high output. Additionally, we found that REBEL generated an entity with no matching textual mention three times in the sample, suggesting that it also hallucinates on rare occasions.

Although PL-Marker had very low output, it generated notably reliable and sensible results. Its success is qualified by the fact that the relations it generates, which are from SciERC and ACE-2005, are much more broadly defined and have fewer syntactic rules than those in REBEL or UniRel. If used in a KE workflow, the resultant KG would be much less precise and informative than one created with more strongly defined relations, such as the Wikidata properties used in REBEL.

Additionally, we report a combined accuracy, which is an unweighted average of syntactic and semantic accuracy, as in (Yang et al. (2021)).

	# Trip Eval'd	Syn Acc	Sem Acc	Con	Combined Acc	# Trip's	% Docs w/ Predicted Trip
REBEL	181	0.86	0.30	0.99	0.58	4766	99
PL-Marker (bert) RE	18	**1.0**	0.94	1.0	0.97	289	10
PL-Marker (scibert) RE	9	**1.0**	0.56	1.0	0.78	127	4
PL-Marker (albert-xxl) RE	4	**1.0**	**1.0**	1.0	**1.0**	147	4
Unirel (NYT)	6	0.17	0.0	1.0	0.083	87	2
DeepStruct (NYT)	5	0.80	0.60	1.0	0.7	93	3

Table 17: Relation Extraction Qualitative Evaluation Results

Since UniRel and PL-Marker's scibert and albert-based models return so few triples on the 100 sample records, we also perform a supplementary evaluation of all the triples predicted over the complete OMIn dataset. These scores can be seen in Table 18.

Lastly, since the PL-Marker's bert-based model returned very few triples in the evaluation sample but too many to evaluate by hand, we selected at random 1000 records from the OMIn dataset and qualitatively evaluated the resulting triples. These scores are available in Table 19.

	# Trip's Eval'd	Syn Acc	Sem Acc	Con	Combined Acc
PL-Marker (albert-xxl) RE	158	**0.98**	**0.98**	1.0	**0.98**
PL-Marker (scibert) RE	127	**1.0**	0.78	1.0	0.89
DeepStruct (NYT)	93	0.79	0.65	1.0	0.72
UniRel (NYT)	87	0.37	0.14	0.98	0.26

Table 18: Supplementary Relation Extraction Evaluation

	# Trip's Eval'd	Syn Acc	Sem Acc	Con	Combined Acc
PL-Marker (bert) RE	126	0.96	0.99	1.0	0.98

Table 19: Supplementary Relation Extraction Evaluation, PL-Marker BERT Model

7 Discussion

7.1 Performance

The selected tools scored significantly lower on the OMIn dataset than on benchmark datasets (in the majority of cases). Some notable exceptions are the CR tools, s2e-coref and ASP, and the RE tools, PL-Marker and REBEL. Both NER and NEL tools failed to reliably extract GS entities. NER tools, in general, performed much better on benchmark-annotated GSs, but still significantly below reported scores for those benchmarks, which indicates that they struggle to transfer to the maintenance domain. Additionally, we found that errors in identifying entity spans often arose in sentences with uncommon syntax and shorthand, acronyms and abbreviations were often ignored or misidentified, and the overall efficacy of knowledge extraction was limited by the prevalence of omitted subjects in sentences.

7.2 Trust

In this work, we focused on trust in four facets:

- **Privacy and Confidentiality** None of the NLP nor LLM models we evaluated were allowed to leverage data storage or APIs external to our private testing infrastructure. This allows an organization to keep their confidential information private.
- **Accuracy and Robustness** Each tool's knowledge extraction capability was evaluated to determine what level of accuracy and organization could expect from an NLP tool or LLM not trained or tuned for their domain. Evaluating tools in diverse domains is important to understand robustness.
- **Reproducibility** It is essential that results from the tools selected can be reproduced and do not vary from test to test. Since we repeatedly evaluated zero-shot scores for 16 different tools, we ensured that results were not influenced by previous data passed into the tools and models.

Further, we evaluated the degree of complexity to build and run the tools as discussed more in the following section.

- **Accountability** We selected and our gold standard dataset and evaluation metrics based on peer reviewed community standards. We then documented all of the processes and procedures either directly in this work, its appendix or the public OMIn data repository.

7.3 Technology Readiness Level

The Technology Readiness Levels (TRLs) provide a 9-level gauge of how close a tool is to being ready for launch. The scale ranges from basic technology research at level 1 to system proven in an operational environment at level 9 (Mankins et al. (1995)). Because TRLs communicate technological maturity in the context of a target operational environment, TRL assessments provide a way for stakeholders in operational environments to shape future research and development. We measured the TRLs of the sixteen tools based on their performance on FAA maintenance data. Because the F1 and accuracy scores are very low, the TRL levels are in the 1-2 range (basic technology research and research to produce feasibility) as shown in Table 20. Some tools required preprocessing and did not let FAA data to be passed in directly as text. Additionally, some tools were either outdated, had complex software version dependencies, or did not have clear documentation explaining how to run the tools. This in turn lowered the TRL level rating for the respective tool. A "Reproducibility Rating" for each tool can be found in our ReadMEs on GitHub, which describes the challenges we encountered during the setup process.

NER		Coref		NEL		RE	
spaCy	2	ASP	1	BLINK	2	REBEL	1
flair	1	coref-mt5	1	spaCy Entity Linker	2	UniRel	1
stanza	2	s2e-coref	1	GENRE	1	DeepStruct	1
nltk	1	neuralcoref	1	ReFinED	2	PL-Marker	2

Table 20: Tool TRL Levels for Zero-Shot on FAA Data

8 Conclusion

We present the Operations and Maintenance Intelligence (OMIn) Dataset, with Version 1 curated initially from raw FAA Accident/Incident data. OMIn is curated for KE in operations and maintenance, featuring textual descriptions of maintenance incidents characterized by mentions of aircraft systems and domain-specific shorthand. We release the gold standards prepared for NER, CR, and NEL as part of OMIn. This baseline expands the portfolio in the aviation operations and maintenance domain, since it offers records on a variety of subject matters, long enough to provide context and valuable information

for extraction. OMIn is the first open-source dataset curated for KE in the operation and maintenance domains. It also contains structured data, such as details of the aircraft, failure codes, and dates. The structured data can be used in future work alongside natural language text to develop an integrated and mutually validating KE approach. While OMIn is currently based on aviation maintenance incident data, this data has qualities common to many sets of records or logs in the operation and maintenance domains, making it a valuable baseline. By publicizing this dataset on our GitHub repository, along with a Zenodo DOI (Mealey et al. (2024)), we offer it to the community and invite collaboration toward a robust, open-source KE dataset for the domain. We will respond to GitHub Issues and release subsequent versions in partnership with the community.

Hand in hand with the OMIn benchmark dataset, we present a comprehensive evaluation of sixteen tools across the four stages of the KE workflow. This evaluation provides insights into their performance on real-world data, limitations, and potential for improvement in the maintenance domain. We identified the challenges faced by these tools when dealing with maintenance-specific data, highlighting the need for domain-specific training and adaptation to enhance their effectiveness and reliability.

Future work can build on this study by improving the OMIn benchmark data quality through pre-processing techniques, enhancing gold standards by adding domain-specific entity and relation labels, and expanding gold standards by annotating additional data. As the state-of-the-art in NLP continues to progress, the OMIn benchmark can be used to evaluate new KE-focused NLP tools like the ones discussed in this paper, as well as LLM-based approaches, such as agentic workflows in which LLMs plan and orchestrate subtasks performed by smaller, specialized models. We hope that the release of our dataset enables the community to explore both the incremental improvement of KE tools and the strategic incorporation of structured knowledge resources to advance the field toward more trustworthy and dynamic knowledge extraction systems for operations and maintenance.

9 Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used the web interfaces of GPT-4o, https://chatgpt.com/, in order to improve the readability and flow of this manuscript. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

10 Acknowledgments

We would like to thank University of Notre Dame students Danny Finch, Alyssa Riter, and Lindsey Michie for their contributions to curating the OMIn baseline

data set. We would also like to thank Crane NSWC Technical Points of Contact (TPOCs) Alicia Scott, Eli Phillips, Aimee Flynn, Adam Shull, and Tim Kelley for their insight into operational priorities and challenges related to trusted MO. Professor Christopher Sweet provided consultation on metric selection and development. The Notre Dame Center for Research Computing and the SCALE Program at Purdue University provided funding support.

References

Akbik, A., Blythe, D., Vollgraf, R., 2018. Contextual string embeddings for sequence labeling, in: COLING 2018, 27th International Conference on Computational Linguistics, pp. 1638–1649.

Akhbardeh, F., Desell, T., Zampieri, M., 2020a. Maintnet: A collaborative open-source library for predictive maintenance language resources. arXiv preprint arXiv:2005.12443 .

Akhbardeh, F., Desell, T., Zampieri, M., 2020b. NLP tools for predictive maintenance records in MaintNet, in: Wong, D., Kiela, D. (Eds.), Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Suzhou, China. pp. 26–32. URL: https://aclanthology.org/2020.aacl-demo.5.

Al-Moslmi, T., Gallofré Ocaña, M., L. Opdahl, A., Veres, C., 2020. Named Entity Extraction for Knowledge Graphs: A Literature Overview. IEEE Access 8, 32862–32881. URL: https://ieeexplore.ieee.org/document/8999622/?arnumber=8999622, doi:10.1109/ACCESS.2020.2973928. conference Name: IEEE Access.

Ameri, F., Tahsin, R., 2022. Knowo: A tool for generation of semantic knowledge graphs from maintenance workorders data, in: Kim, D.Y., von Cieminski, G., Romero, D. (Eds.), Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas into Action, Springer Nature Switzerland, Cham. pp. 188–195.

Ayoola, T., Tyagi, S., Fisher, J., Christodoulopoulos, C., Pierleoni, A., . ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking, in: Loukina, A., Gangadharaiah, R., Min, B. (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, Association for Computational Linguistics. pp. 209–220. URL: https://aclanthology.org/2022.naacl-industry.24, doi:10.18653/v1/2022.naacl-industry.24.

Bagga, A., Baldwin, B., 1998. Algorithms for scoring coreference chains, in: The first international conference on language resources and evaluation workshop on linguistics coreference, Citeseer. pp. 563–566.

Batista, D., 2018. Named-Entity evaluation metrics based on entity-level. URL: https://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/.

Bohnet, B., Alberti, C., Collins, M., 2022. Coreference Resolution through a seq2seq Transition-Based System. URL: http://arxiv.org/abs/2211.12142, arXiv:2211.12142.

Bratanic, T., 2021. From text to knowledge: The information extraction pipeline URL: https://towardsdatascience.com/from-text-to-knowledge-the-information-extraction-pipeline.

Brundage, M.P., Sexton, T., Hodkiewicz, M., Dima, A., Lukens, S., 2021. Technical language processing: Unlocking maintenance knowledge. Manufacturing Letters 27, 42–46.

Consortium, L.D., 2000. Entity detection and tracking – phase 1: Ace pilot study task definition. Linguistic Data Consortium. URL: https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/edt-phase1-v2.2.pdf.

Consortium, L.D., 2002. Entity detection and tracking - phase 1: Edt and metonymy annotation guidelines. Linguistic Data Consortium. URL: https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/edt-guidelines-v2-5.pdf.

De Cao, N., Izacard, G., Riedel, S., Petroni, F., . Autoregressive Entity Retrieval. URL: http://arxiv.org/abs/2010.00904, arXiv:2010.00904.

Dhaini, M., Hussain, K.Z., Zaradoukas, E., Kasneci, G., 2025. Evalxnlp: A framework for benchmarking post-hoc explainability methods on nlp models. arXiv preprint arXiv:2505.01238 .

Dixit, S., Mulwad, V., Saxena, A., 2021. Extracting semantics from maintenance records. ArXiv abs/2108.05454. URL: https://api.semanticscholar.org/CorpusID:236986887.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Larson, J., 2024. From local to global: A graph rag approach to query-focused summarization. URL: https://arxiv.org/abs/2404.16130, arXiv:2404.16130.

Federal Aviation Administration, 2024. FAA Accident and Incident Data System. https://www.asias.faa.gov/apex/f?p=100:189:::NO.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Kirrane, S., Labra Gayo, J.E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.C., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J.F., Staab, S., Zimmermann, A., 2021. Quality Assessment. Springer. Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge, pp. 119–125. URL: https://kgbook.org/, doi:10.2200/S01125ED1V01Y202109DSK022.

Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A., 2020. spacy: Industrial-strength natural language processing in python doi:10.5281/zenodo.1212303.

Huguet Cabot, P.L., Navigli, R., 2021. REBEL: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic. pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204.

Jiang, J.J., Conrath, D.W., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. URL: https://arxiv.org/abs/cmp-lg/9709008, arXiv:cmp-lg/9709008.

Khorashadizadeh, H., Amara, F.Z., Ezzabady, M., Ieng, F., Tiwari, S., Mihindukulasooriya, N., Groppe, J., Sahri, S., Benamara, F., Groppe, S., 2024. Research trends for the interplay between large language models and knowledge graphs. arXiv preprint arXiv:2406.08223 .

Kirstain, Y., Ram, O., Levy, O., 2021. Coreference Resolution without Span Representations, in: Zong, C., Xia, F., Li, W., Navigli, R. (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics. pp. 14–19. URL: https://aclanthology.org/2021.acl-short.3, doi:10.18653/v1/2021.acl-short.3.

Konys, A., 2018. Towards knowledge handling in ontology-based information extraction systems. Procedia Computer Science 126, 2208–2218. URL: https://www.sciencedirect.com/science/article/pii/S1877050918312031, doi:https://doi.org/10.1016/j.procs.2018.07.228. knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.

Levesque, H., Davis, E., Morgenstern, L., 2012. The winograd schema challenge, in: Thirteenth International Conference on Principles of Knowledge Representation and Reasoning.

Li, Z., Wang, Y., Fan, R., Wang, Y., Li, J., Wang, S., 2024. Learning to adapt to low-resource paraphrase generation. arXiv preprint arXiv:2412.17111 .

Liu, B., Li, X., 2025. Large language models for knowledge graph embedding techniques, methods, and challenges: A survey. arXiv preprint arXiv:2501.07766 .

Liu, T., Jiang, Y., Monath, N., Cotterell, R., Sachan, M., . Autoregressive Structured Prediction with Language Models. URL: http://arxiv.org/abs/2210.14698, arXiv:2210.14698.

Liu, Y., Hou, J., Chen, Y., Jin, J., Wang, W., 2025. Llm-acnc: Aerospace requirement texts knowledge graph construction utilizing large language model. Aerospace 12, 463.

Luo, X., 2005. On coreference resolution performance metrics, in: Proceedings of human language technology conference and conference on empirical methods in natural language processing, pp. 25–32.

Mankins, J.C., et al., 1995. Technology readiness levels. White Paper, April 6, 1995.

Matviishyn, O., 2025. How to use large language models (llms) with enterprise and sensitive data. StartupSoft blog. URL: https://www.startupsoft.com/llm-sensitive-data-best-practices-guide/.

Mealey, K., Karr, J., Saboia Moreira, P., , , Brenner, P., Vardeman, C., 2024. Operations and Maintenance Intelligence (OMIn) Dataset. URL: https://zenodo.org/doi/10.5281/zenodo.13333824, doi:10.5281/zenodo.13333824.

Meng, X., Jing, B., Wang, S., Pan, J., Huang, Y., Jiao, X., 2023. Fault knowledge graph construction and platform development for aircraft phm. Sensors 24, 231.

Meta, . Papers with Code - The latest in Machine Learning. URL: https://paperswithcode.com/.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., The Annotation Group, 1998. BBN: Description of the SIFT System as Used for MUC-7, in: Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998. URL: https://aclanthology.org/M98-1009.

Mishra, B.D., Tandon, N., Clark, P., 2017. Domain-Targeted, High Precision Knowledge Extraction. Transactions of the Association for Computational Linguistics 5, 233–246. URL: https://doi.org/10.1162/tacl_a_00058, doi:10.1162/tacl_a_00058. _eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00058/1567478/tacl_a_00058.pdf.

Moosavi, N.S., Strube, M., 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric, in: Proceedings of the 54th annual meeting of the association for computational linguistics, Association for Computational Linguistics. pp. 632–642.

NASA Aviation Safety Reporting System, . NASA ASRS Dataset. https://asrs.arc.nasa.gov/search/database.html.

NASA Prognostics Center of Excellence, 2023. NASA Prognostics Center of Excellence. https://www.nasa.gov/content/nasa-prognostics-center-of-excellence.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y., 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes, in: Pradhan, S., Moschitti, A., Xue, N. (Eds.), Joint Conference on EMNLP and CoNLL - Shared Task, Association for Computational Linguistics, Jeju Island, Korea. pp. 1–40. URL: https://aclanthology.org/W12-4501.

Segura-Bedmar, I., Martínez, P., Herrero-Zazo, M., 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013), in: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp. 341–350.

Shan, A., Bauer, J., Carlson, R., Manning, C., 2023. Do "English" Named Entity Recognizers Work Well on Global Englishes?, in: Bouamor, H., Pino, J., Bali, K. (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore. pp. 11778–11791. URL: https://aclanthology.org/2023.findings-emnlp.788, doi:10.18653/v1/2023.findings-emnlp.788.

Shanmugarasa, Y., Ding, M., Chamikara, M., Rakotoarivelo, T., 2025. Sok: The privacy paradox of large language models: Advancements, privacy risks, and mitigation. arXiv preprint arXiv:2506.12699 .

Sharp, M., Sexton, T., Brundage, M.P., 2017. Toward semi-autonomous information, in: Lödding, H., Riedel, R., Thoben, K.D., von Cieminski, G., Kiritsis, D. (Eds.), Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing, Springer International Publishing, Cham. pp. 425–432.

Shen, W., Li, Y., Liu, Y., Han, J., Wang, J., Yuan, X., 2021. Entity linking meets deep learning: Techniques and solutions. IEEE Transactions on Knowledge and Data Engineering 35, 2556–2578.

Sukthanker, R., Poria, S., Cambria, E., Thirunavukarasu, R., 2020. Anaphora and coreference resolution: A review. Information Fusion 59, 139–162.

Sundheim, B.M., 1995. Overview of Results of the MUC-6 Evaluation, in: Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995. URL: https://aclanthology.org/M95-1002.

Tang, W., Xu, B., Zhao, Y., Mao, Z., Liu, Y., Liao, Y., Xie, H., . UniRel: Unified Representation and Interaction for Joint Relational Triple Extraction. URL: http://arxiv.org/abs/2211.09039, arXiv:2211.09039.

Usbeck, R., Röder, M., Ngonga Ngomo, A.C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., et al., 2015. Gerbil: general entity annotator benchmarking framework, in: Proceedings of the 24th international conference on World Wide Web, pp. 1133–1143.

USC-ISI-I2, 2021. Kgtk similarity. https://github.com/usc-isi-i2/kgtk-similarity.

Vilain, M., Burger, J.D., Aberdeen, J., Connolly, D., Hirschman, L., 1995. A model-theoretic coreference scoring scheme, in: Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995.

Wang, C., Liu, X., Chen, Z., Hong, H., Tang, J., Song, D., . DeepStruct: Pretraining of Language Models for Structure Prediction. URL: http://arxiv.org/abs/2205.10475, arXiv:2205.10475.

Webster, K., Costa-jussà, M.R., Hardmeier, C., Radford, W., 2019. Gendered Ambiguous Pronoun (GAP) Shared Task at the Gender Bias in NLP Workshop 2019, in: Costa-jussà, M.R., Hardmeier, C., Radford, W., Webster, K. (Eds.), Proceedings of the First Workshop on Gender Bias in Natural Language Processing, Association for Computational Linguistics, Florence, Italy. pp. 1–7. URL: https://aclanthology.org/W19-3801, doi:10.18653/v1/W19-3801.

Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L., 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. URL: http://arxiv.org/abs/1911.03814, arXiv:1911.03814.

Wu, Z., Guo, J., Hou, J., He, B., Fan, L., Yang, Q., 2024. Model-based differentially private knowledge transfer for large language models. arXiv preprint arXiv:2410.10481 .

Yang, J., Li, Y., Gao, C., Zhang, Y., 2021. Measuring the short text similarity based on semantic and syntactic information. Future Generation Computer Systems 114, 169–180.

Ye, D., Lin, Y., Li, P., Sun, M., 2022. Packed levitated marker for entity and relation extraction, in: Muresan, S., Nakov, P., Villavicencio, A. (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, Association for Computational Linguistics. pp. 4904–4917. URL: https://aclanthology.org/2022.acl-long.337.

Yu, H., Li, H., Mao, D., Cai, Q., 2020. A relationship extraction method for domain knowledge graph construction. World Wide Web 23, 735–753. URL: https://doi.org/10.1007/s11280-019-00765-y, doi:10.1007/s11280-019-00765-y.

Yue, S., Xiao, L., Li, J., Wang, N., 2022. Research on application of knowledge graph for aircraft maintenance. Advances in Mechanical Engineering 14, 16878132221107429. URL: https://doi.org/10.1177/16878132221107429, doi:10.1177/16878132221107429, arXiv:https://doi.org/10.1177/16878132221107429.

A CoNLL-2012 Format Pre-Processing

For CoNLL-2012, documents were organized in a tabular structure with each word on a separate line and more than 11 columns detailing the word's semantic role. These columns include parts of speech, parse bits, predicate lemmas, speakers, and named entities. The parse bit links each word to a segment of the sentence's Propbank-style parse tree, originally generated using a Charniak parser by the CoNLL-2012 developers. To recreate their process, we used the Charniak parser from Brown Laboratory for Linguistic Information Processing (BLLIP), along with the NLTK POS-tagger and spaCy EntityRecognizer. The CoNLL-2012 developers used the Identifinder^{TM} tool from BBN (Miller et al. (1998)) for NER; however, we used spaCy since it was more up-to-date. All records were uniformly labeled with 'speaker1' in the speaker field. We found through experimentation that the selected CoNLL-2012-based tools did not use the predicate-lemma and its associated columns, so we inserted placeholders in these fields. The OMIn data formatted for CoNLL-2012 is available in our repository (Mealey et al. (2024)).

Although we aimed to replicate the original CoNLL-2012 dataset from OntoNotes 5.0, discrepancies in the POS-tagging, NER, and parsing tools led to a less-than-perfect match. Without access to the original tools, our version serves as a close approximation.

B NER Annotation Guidelines

We follow ACE-2005 to label persons, locations, organizations, geo-political entities (GPEs), facilities, weapons, and vehicles. We make a few exceptions. First, we include "ground" and "land" as location entities, since they are distinct locations in aviation, where they are often used to differentiate from airspace. We also exclude articles from our entities, but keep all other modifiers. Lastly, we do not include relative clauses or relative pronouns in our GS, since they are unhelpful as a basis for NEL.

We follow OntoNotes 5.0 to label dates, times, quantities, and cardinals.

Additionally, we label entities that fall into one of the following categories: vehicle system/component, operational items (fuel, oil, load, etc.), failures, causes of failures, symptoms of failures, phases of flight (takeoff, climb, landing, etc.), types of flight (ferry flight, test flight, etc.), and procedures (maintenance, safety checks, etc.). Future work could involve formalizing these categories into well-defined entity types. We follow ACE-2005 in all syntactical rules such as the inclusion of modifying phrases (except for articles), nesting entities, treating appositives, etc.

We ignore all typos and words which are cut off at the end of the record. However, we include shorthand and acronyms ("acft" for aircraft, "prop" for propeller, etc.).

Example 1: Nested Entities For the record, "(-23) Mr. Timothy Allen Wells was acting as pilot in command of a Bell Helicopter model BHT-47-G5, N4754R, engaged", we follow ACE-2005's nested entity guidelines and include "Mr. Timothy Allen Wells", "pilot in command of a Bell Helicopter model BHT-47-G5, N4754R", "Bell Helicopter model BHT-47-G5, N4754R", "Bell Helicopter model BHT-47-G5", "Bell", and "N4754R".

Example 2: Aviation Entities For the record, "After departing high oil temp. Landed off airpor. Sheared main gear. Found low on oil.", the entities are "high oil temp","oil","sheared main gear", and "OIL". The first "OIL" is an entity nested in "high oil temp". "oil" is included because it is an operational item, "high oil temp" is included because it is a failure or a failure symptom, and "sheared main gear"" is both a failure and a system component. Since we do not label the entities with definite types, the ambiguity of "sheared main gear"'s type is not an issue.

C NEL Evaluation Guidelines

C.1 F1 Score Details

We define a true positive as a predicted entity that matches both the entity and the QID in a gold standard link. A false positive is a predicted entity that matches an entity but not its QID in a gold standard link. A false negative exists when there is no matching predicted entity for a gold standard entity-QID link. Predicted entities without any QID, as well as predicted entities-QID links without a matching gold entity, are not included in the evaluation.

C.2 OT Metrics Details

For the OT metrics, we evaluate the intersection of entities in the gold and predicted sets. This intersection consists of predicted entities that have a matching entity in the gold standard, where both predicted and gold entities are linked with a QID. We then report a micro-average across all linked entities in the intersection, excluding those for which a score could not be obtained due to limitations in the KB.

Jiang Conrath (JC) is an information-theoretic distance metric that combines path-based features with information content to provide a nuanced similarity metric Jiang and Conrath (1997). The formula is given by:

$$\mathrm{jc}(c_1, c_2) = 2 \cdot \log p(\mathrm{mss}(c_1, c_2)) - (\log p(c_1) + \log p(c_2)) \tag{1}$$

where $\mathrm{jc}(c_1, c_2)$ denotes the distance between concepts c_1 and c_2, $\mathrm{mss}(c_1, c_2)$ is the most specific subsumer of c_1 and c_2, and $p(c)$ is the probability of encountering an instance of concept c. In USC-ISI-I2 (2021), they use instance counts of a class to compute the probability $p(c)$ and normalize Jiang Conrath distance

onto a $[0 \dots 1]$ similarity measure by dividing by the largest possible distance between c_1 and c_2 through the *entity* node (Q351201) in the ontology.

The *class* similarity computation employs the Jaccard Similarity of the superclass sets of two nodes, inversely weighted by the instance counts of the classes. Formally, for two concepts c_1 and c_2, let $S(c_1)$ and $S(c_2)$ represent their respective sets of superclasses, and let $I(c)$ denote the instance count of class c. The class similarity is defined as:

$$\text{class\_sim}(c_1, c_2) = \frac{\sum_{c \in S(c_1) \cap S(c_2)} \frac{1}{I(c)}}{\sum_{c \in S(c_1) \cup S(c_2)} \frac{1}{I(c)}} \tag{2}$$

where the term $\frac{1}{I(c)}$ inversely weights each class by its instance count, thereby reducing the influence of more general classes with higher instance counts and emphasizing more specific classes.

C.3 Flexible Evaluation

Flexible evaluation makes use of the Flexible NEL GS, as described in Section 4.3.3. The Flexible GS includes secondary and tertiary linked entities as well as the primary linked entities used in the other evaluation strategies. In Flexible evaluation, if a predicted linked entity exactly matches either the primary, secondary, or tertiary link, it is correct. Flexible evaluation utilizes strong-matching.

Example: "Forward cargo door opened as aircraft took off. Objects dropped out. Returned. Failed to see warning light."

The primary, secondary, and tertiary entities are laid out in Table 21. In strong and weak-matching evaluation, only ("aircraft",Q11436) would be included in the gold standard. In flexible evaluation, ("door", Q36794), ("aircraft",Q11436), and ("light", Q1146001) would be included.

Primary Ent	**Primary QID**	**Secondary Ent**	**Secondary QID**	**Tertiary Ent**	**Tertiary QID**
forward cargo door		cargo door		door	Q36794
aircraft	Q11436				
warning light		light	Q1146001		

Table 21: Example NEL Flexible Entities and QIDs

Secondary entities are also linked to primary QIDs when available, and so too with tertiary entities to secondary and primary QIDs. This is done so that if a tool links a more "general" mention to the QID for the fitting, context-specific Wikidata entity, rather than the general QID, it is not penalized. For example, the GS for a document in the FAA data includes the primary link ("forced landing", Q1975745) and the secondary link ("landing", Q844947). If a tool predicted ("landing",Q1975745), that would be counted as correct, since it inferred from context that it was a forced landing and linked it to the corresponding QID.

D RE Evaluation Guidelines

D.1 Syntactic Accuracy

Syntactic accuracy is the degree to which a tool's output follows the grammatical rules in our set of guidelines. A triple is either completely syntactically accurate (1.0), half syntactically accurate (0.5), or syntactically inaccurate (0.0), depending on whether both, one of, or neither of the head and tail entities are correct, respectively. Our guidelines are recorded below:

- Head and tail entities must consist of complete phrases. "Complete phrase" signifies a word or phrase which can be treated as a noun or a verb. For example, the triple ("cowling","part of","engine in") is inaccurate, since "engine in" is not a complete phrase.
- If a word or phrase is used as a modifier in a sentence (and is thus not its own phrase in that particular sentence), it may still be counted as a complete phrase if it can function as a noun, verb, noun phrase, or verb phrase in another context. For example, in the sentence "Wing fuel tank sumps were not drained during preflight", ("sumps","part of","wing fuel tank") and ("fuel tank sumps", "part of", "wing") would both be syntactically accurate.
- **Exception** to the above two rules: personal pronouns may be entities, and should be interpreted the same as the person they refer to.
- If a head or tail entity includes words or phrases that modify a part of the sentence outside of that included in the entity, it is inaccurate. For example, in the sentence "Engine cowling separated from engine in flight," the subject is "engine cowling", and the verb is "separated", modified by "in flight" and "from engine." Because "in flight" modifies "separated," the predicted entity "engine in flight" would be syntactically inaccurate.
- Verbs and verb phrases may only be used as entities if the relation can accept an event-type entity. Verb phrases also do not need to have a subject. For example: ("improper preflight", "has effect", "crashed") is syntactically accurate.
- Complete clauses (subject-verb) may only be used as entities if the relation can accept an event-type entity.
- A head entity may be a subspan of its tail entity, and vice versa.
- Head and tail entities must follow syntax constraints implied by the relation. For example, the relation "place of birth" must have a location as the head and a person as the tail. These constraints are described for each set of relations below under Syntax Constraints.

D.2 Syntax Constraints

D.2.1 NYT Relation Set

NYT is used by the NYT models in UniRel and DeepStruct.

There are 25 possible relations. The relations which most commonly occur in UniRel and DeepStruct output on FAA data are "/location/location/contains", "/location/neighborhood/neighborhood_of", "/business/person/company", and "/business/company/place_founded".

NYT relations are made up of three parts: the head entity is the subject, the middle part is a category for the head entity and the third part is a category or verb phrase that defines the relationship of the tail entity to the head. All triples where the head does not belong to the category in the middle part of the relation are docked 0.5 in syntactic accuracy. Similarly, if the last part of the relation is a category, such as "place founded", then if the tail does not correspond to that category, i.e., is not a place, it is docked 0.5. Some relations have a verb phrase as the relation instead, such as "contains". In this case, the tail must follow any implicit constraints of that verb. For example, the tail for "contains" cannot be an abstract concept, since a location cannot contain an abstract concept. To summarize in an example: the relation /business/company/advisors must have a company as the head entity and a group of people as the tail.

D.2.2 ACE-2005 Relation Set

ACE-2005 is used by PL-Marker and DeepStruct. The ACE-2005 relations and their constraints on head and tail entities are laid out in Table 22.

Relation	**Head**	**Tail**
PER-SOC	Person(s)	Person(s)
ART	Person(s)	Physical Object
ORG-AFF	Person(s)	Organization
GEN-AFF	Any	Any
PHYS	Person(s)	Anything with a physical form
PART-WHOLE	Category must correspond to Tail	Category must correspond to Head

Table 22: ACE-2005 Relations' Syntax Constraints on Head and Tail Entities

D.2.3 SciERC Relation Set

SciERC is used by PL-Marker. The SciERC relations and their constraints are laid out in Table 23.

Relation	Head	Tail
PART-OF	Category must correspond to Tail	Category must correspond to Head
USED-FOR	Any	Any
FEATURE-OF	Category must correspond to Tail	Category must correspond to Head
CONJUNCTION	Category must correspond to Tail	Category must correspond to Head
HYPONYM-OF	Category must correspond to Tail	Category must correspond to Head
COMPARE	Category must correspond to Tail	Category must correspond to Head

Table 23: SciERC Relations' Syntax Constraints on Head and Tail Entities

D.2.4 REBEL Wikidata Property Relation Set

REBEL utilizes a curated set of 220 relations derived from Wikidata Properties. The complete set is available in the REBEL repository.[9] The top 10 relations appearing in REBEL's output on the FAA data are: has part, part of, different from, subclass of, instance of, has effect, has cause, located in the administrative territorial entity, product or material produced, and facet of.

For all REBEL relations (Wikidata properties), the head and tail entities must match usage in Wikidata. This is judged by the evaluator. Some notable properties are:

- "different from" is only used when head and tail entities share a similar name. The "different from" relation is used to distinguish entities named the same way or similarly enough that they need to be distinguished.
- "has effect" and "has cause" may have noun phrases or verb phrases on either end
- "has part", "part of", "subclass of", "instance of", and "facet of" all imply that the head and tail entities must correspond in entity category. We refer to coarsely defined categories such as event, physical object, time, quantity, date, and abstract concept, which are obvious to the annotator. If a categorical difference between head and tail is possible but not obvious, we count it as syntactically correct.

D.3 Semantic Accuracy

Semantic accuracy is the degree to which the tool's output adheres to the real world. A triple is either semantically accurate (1.0) or semantically inaccurate (0.0). The guidelines are recorded below:

- The evaluator is encouraged to use their domain expertise as well as all outside knowledge available.
- If a head or tail entity is an incomplete phrase or includes extraneous words, the triple will still be counted as semantically accurate if using subspans of those entities enables a sensible triple. For example, in the record,

[9]The complete set of relations can be accessed in the REBEL repository at: https://raw.githubusercontent.com/Babelscape/rebel/main/data/relations_count.tsv.

"Engine ran rough. Pilot landed in field," if the triple were ("engine", "used by", "pilot landed") were given, it would be counted as semantically accurate, since ("engine","used by","pilot") is accurate.

D.4 Consistency

Consistency is the degree to which the set of output triples for each record/document is free of contradictions. Percent consistency is calculated via the expression: $(N_{triples}$ - $N_{inconsistencies})/(N_{triples})$, where $N_{inconsistencies}$ is the number of triples such that if they were removed from the set of output triples, the remaining set would be consistent. For example, if there are 3 triples generated for a document and 2 of them contradict each other, there is 1 inconsistency since if one of the contradicting triples were removed, the remaining 2 would be consistent. In this case, it would receive a consistency score of 0.6667.

- An example of contradicting triples would be ("Brookline, MA","place of birth","John F. Kennedy") and ("John F. Kennedy","has place of birth","Boston, MA")
- Most relations do not necessitate a one-to-one relation, however. In the record, "Crashed when load wedged in trees. Improper preflight," if the triples ("improper preflight", "has effect", "crashed") and ("crashed","has cause","load wedged") were generated, this would still be consistent, since an event may have multiple causes.