TU WIEN Informatics

Extracting a Temporal Supply Chain Knowledge Graph from a Large-Scale News Corpus

Evidence from Ukraine's Drone Industry

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Maximilian Laurent Heß, B.Sc.
Matrikelnummer 12347554

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Allan Hanbury
Mitwirkung: Christian Diem, PhD

Wien, 30. August 2026

Maximilian Laurent Heß	Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Informatics

Extracting a Temporal Supply Chain Knowledge Graph from a Large-Scale News Corpus

Evidence from Ukraine's Drone Industry

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Maximilian Laurent Heß, B.Sc.
Registration Number 12347554

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Allan Hanbury
Assistance: Christian Diem, PhD

Vienna, August 30, 2026

Maximilian Laurent Heß	Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Erklärung zur Verfassung der Arbeit

Maximilian Laurent Heß, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 30. August 2026

Maximilian Laurent Heß

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung

Zunächst möchte ich mich ganz herzlich bei meinem Betreuer Allan Hanbury bedanken, der von Anfang an von der Idee dieser Arbeit überzeugt war, mich bei Fragen stets unterstützt und mir mit seinem Feedback sehr geholfen hat.

Besonders bedanken möchte ich mich bei Christian Diem, mit dem ich unzählige Diskussionen über Lieferkettendefinitionen und Annotationsergebnisse führen durfte und der mir stets zur Seite stand und mir immer Feedback gab, wenn ich es brauchte. Nie vergessen werde ich seine Einladung nach Oxford, die es mir ermöglichte, diese Arbeit dort als Visiting Student voranzutreiben. Danke, Christian!

Als Nächstes möchte ich mich beim Supply Chain Intelligence Institute Austria (ASCII) bedanken, bei dem ich während meines gesamten Masterstudiums als Research Assistant tätig war und auch diese Arbeit verfassen durfte. Besonderer Dank gilt hier Peter Klimek für sein stets herausragendes Feedback und Georg Heiler, ohne den ich wahrscheinlich weder Data Science studiert noch beim ASCII gearbeitet hätte.

Zuletzt möchte ich mich bei meiner Familie und meiner Freundin von ganzem Herzen bedanken. Danke, Barbara und Oliver, danke, Lukas und Mila, danke, Dieter und Marlis, danke, Gabriel, danke, Ruth und Rita, danke, Carmen und Patrick und danke, Anna Lena! Ohne eure Unterstützung und Liebe wäre diese Thesis gar nicht erst möglich gewesen!

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acknowledgements

First of all, I would like to express my sincere thanks to my supervisor, Allan Hanbury, who was convinced of the idea behind this thesis from the very beginning, always supported me whenever I had questions, and was a great help to me with his feedback.

I would particularly like to thank Christian Diem, with whom I had the opportunity to have countless discussions about supply chain definitions and annotation results, and who was always there for me and provided feedback whenever I needed it. I will never forget his invitation to Oxford, which enabled me to make progress on this thesis there as a visiting student. Thank you, Christian!

Next, I would like to thank the Supply Chain Intelligence Institute Austria (ASCII), where I worked as a research assistant throughout my Master's degree and was also able to write this thesis. Special thanks go to Peter Klimek for his consistently outstanding feedback and to Georg Heiler, without whom I would probably neither have studied Data Science nor worked at ASCII.

Finally, I would like to thank my family and my girlfriend from the bottom of my heart. Thank you, Barbara and Oliver, thank you, Lukas and Mila, thank you, Dieter and Marlis, thank you, Gabriel, thank you, Ruth and Rita, thank you, Carmen and Patrick, and thank you, Anna Lena! Without your support and love, this thesis would never have been possible!

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Um Lieferkettenrisiken und Disruptionen zu verstehen, ist es essenziell zu wissen, wie Firmen in einer Lieferkette miteinander verbunden sind und wie sich diese Beziehungen über die Zeit verändern. Ein Lieferkettengraph stellt diese Verbindungen durch Knoten und Kanten dar, die über die Zeit neu erscheinen und wieder verschwinden. Bestehende Methoden zur Konstruktion eines solchen Lieferkettengraphen sind jedoch dadurch limitiert, dass sie keine zeitliche Dimension darstellen, auf proprietären oder nicht öffentlich zugänglichen Daten beruhen oder nur mit Daten von Firmenwebsites arbeiten können.

Hier knüpft diese Thesis an und erforscht anhand der ukrainischen Drohnenindustrie, welche zeitlichen Entwicklungen ein aus einem frei zugänglichen Nachrichtenkorpus konstruierter temporaler Lieferkettengraph abbilden kann. Damit wird das Potenzial von Nachrichtenartikeln als Datengrundlage für die Konstruktion temporaler Lieferkettengraphen untersucht. Darüber hinaus analysiert die Thesis, welche Anteile der extrahierten Kanten im Graphen aus ukrainischen und englischen Quellen stammen, um zu bewerten, wie wichtig Nachrichten in mehreren Sprachen für die Konstruktion des Lieferkettengraphen sind. Dieser Aspekt ist in ähnlichen Arbeiten bisher wenig erforscht.

Zu diesem Zweck entwickelt diese Thesis eine mehrstufige Pipeline, die relevante ukrainisch- und englischsprachige Nachrichtenartikel aus dem Common Crawl News Repository abruft und daraus Beziehungen wie Lieferanten-Käufer- oder Produzenten-Produkt-Beziehungen extrahiert. Aus diesen Beziehungen konstruiert die Pipeline anschließend den temporalen Lieferkettengraphen.

Die Ergebnisse der Thesis zeigen, dass der konstruierte Graph das Wachstum der ukrainischen Drohnenlieferkette durch eine steigende Anzahl von Beziehungen zwischen Firmen sowie zwischen Firmen und deren Produkten widerspiegelt. Zudem bildet der Graph einzelne Entwicklungen wie die Entstehung neuer Drohnentypen ab, wobei nicht alle zeitlichen Entwicklungen korrekt dargestellt werden. Die Analyse der Quellsprachen zeigt außerdem, dass ohne ukrainischsprachige Nachrichtenartikel mehr als 60% der Kanten im Graphen nicht abgebildet wären.

Insgesamt deuten diese Ergebnisse darauf hin, dass generelle zeitliche Trends in einem temporalen Lieferkettengraphen, der aus Nachrichtenartikeln konstruiert wurde, abgebildet werden können. Gleichzeitig verdeutlichen sie, wie wichtig es ist, Nachrichtenartikel in mehreren Sprachen bei der Konstruktion zu berücksichtigen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

Understanding how companies are connected in a supply chain and how these relationships evolve over time is essential for assessing risks and understanding shocks and supply disruptions. A temporal supply-chain graph formally represents these evolving relationships between companies through appearing and disappearing nodes and edges and provides the basis for analysing the supply chain. However, existing methods for constructing such graphs either lack a temporal dimension, rely on proprietary or non-public data sources, or are limited to constructing these graphs from company websites.

In this context, this thesis uses the Ukrainian drone industry as a case study to explore the potential of using news articles as input for temporal supply-chain graph construction by investigating which temporal developments are reflected in a graph constructed from a large-scale news corpus. Additionally, this thesis examines the coverage of relationships extracted from English- and Ukrainian-language sources in the graph to assess the importance of multilingual data sources, an aspect that remains underexplored in comparable works.

For this purpose, this thesis develops a multistep pipeline that retrieves Ukrainian- and English-language news articles from the Common Crawl News repository and extracts relationships from them, including supplier-buyer and maker-product relationships. The extracted relationships are then used to construct a temporal supply-chain graph.

We found that the graph reflects the growth of the Ukrainian drone industry through an increasing number of relationships over time and represents the emergence of new producers of certain drone types, although not all temporal developments are accurately captured. Beyond the temporal analysis, the source-language analysis shows that more than 60% of the relationships would be missing if Ukrainian-language articles were omitted.

Together, these results indicate that general temporal trends can be captured when constructing a temporal supply-chain graph from news articles and demonstrate the importance of including news articles in multiple languages in its construction.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

Kurzfassung **xi**

Abstract **xiii**

Contents **xv**

1 Introduction **1**
1.1 Research Questions and Approach 2
1.2 Related Work 3
1.3 Structure 5

2 Background **7**
2.1 Large-Scale Corpus Deduplication 7
2.2 Inter-Annotator Agreement Metrics 10
2.3 Damerau–Levenshtein 11

3 Data Collection and Corpus Construction **13**
3.1 Common Crawl News 13
3.2 Seed Nodes 14
3.3 Seed Node Filtering and Quality Filter 15
3.4 Deduplication 17
3.5 Composition of the Preprocessed Corpus 18

4 Knowledge-Graph Construction Pipeline **21**
4.1 Document Retrieval 23
4.2 Relevance Classification 29
4.3 Relevant Segment Selection 30
4.4 Named Entity Recognition 33
4.5 Relationship Extraction 33
4.6 Knowledge Graph Construction 37

5 Pipeline Evaluation Design **41**
5.1 Document Retrieval 41
5.2 Relevance Classification 48

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

5.3 Relevant Segment Selection 49
5.4 Named Entity Recognition 50
5.5 Relationship Extraction 51

6 Results **57**
6.1 Document Retrieval 57
6.2 Relevance Classification 60
6.3 Relevant Segment Selection 61
6.4 Named Entity Recognition 63
6.5 Relationship Extraction 68
6.6 Knowledge Graph 72

7 Discussion and Conclusion **79**
7.1 Knowledge-Graph Extraction Pipeline 79
7.2 Answers to the Research Questions 81
7.3 Limitations 82
7.4 Future Work 83
7.5 Conclusion 83

Overview of Generative AI Tools Used **85**

List of Figures **87**

List of Tables **89**

List of Algorithms **93**

Bibliography **95**

Appendix **101**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 1

Introduction

Globally interconnected supply chains are the backbone of modern society, enabling the production and delivery of food, medicines, and other everyday necessities. Understanding their structure is important for assessing the resilience of supply chains and determining how potential shocks propagate through a supply chain and affect the supply of goods [1]. Supply-chain graphs represent this structure by capturing the companies involved as nodes and their relationships, such as supply or ownership relationships, as edges. More sophisticated graphs also include other node types, such as products, and their relationships to companies. However, constructing these graphs is often constrained by data availability [2]. Supply chains are also constantly evolving, with companies entering and exiting supply chains [3]. A supply-chain graph intended to represent these changes therefore requires a temporal dimension.

Existing approaches have used value-added tax (VAT) data to construct national supply-chain graphs for Ecuador [4], Belgium [5], and Hungary [3]. Beyond VAT data, another approach reconstructed a national supply-chain graph using mobile-phone data [6]. However, the underlying VAT and mobile-phone datasets are not publicly available and are focused on national economies, limiting the suitability of approaches based on such datasets for understanding global supply chains. Commercial datasets such as FactSet provide another option, offering cross-country relationships between companies, but they require a subscription and offer incomplete coverage [7].

As an alternative, supply-chain relationships can be extracted from publicly available text [8], [9], [10], [11], [12], [13], such as news articles, blogs, social media, and earnings-call transcripts. While these approaches demonstrate the feasibility of extracting supply-chain relationships from public text, the resulting representations do not have a temporal dimension. One recent approach addresses this gap by constructing a temporal firm network from company websites [7]. However, this leaves open whether temporal supply-chain graphs can also be constructed from other sources, such as news articles.

A further open question concerns the importance of multilingual text sources for supply-chain graph construction. Although such sources have been used in supply-chain extraction [11], to the best of our knowledge, existing work in the supply-chain domain does not separately examine the contribution of individual source languages.

This thesis addresses these gaps by extracting supply-chain relationships involving Ukrainian drone manufacturers, such as supplier-buyer and maker-product relationships, from English- and Ukrainian-language news articles retrieved from a large-scale multilingual news repository. The extracted relationships are used to reconstruct the first-tier supply chains for these manufacturers and are represented in a temporal, multi-relational knowledge graph. The Ukrainian drone industry was selected as a use case because the substantial growth of the sector since the start of the full-scale invasion [14] makes it suitable for investigating whether temporal changes can be captured in the constructed knowledge graph. Moreover, constructing the knowledge graph from Ukrainian- and English-language articles allows us to compare the contribution of each language to the resulting graph.

1.1 Research Questions and Approach

To investigate the temporal reconstruction of the Ukrainian drone supply chain and the contribution of English- and Ukrainian-language sources, this thesis addresses the following research questions.

RQ1: Which temporal trends in the growth of Ukraine's UAV supply chain since the start of the full-scale invasion are reflected in the knowledge graph constructed from multilingual news articles?

RQ2: What difference in relationship coverage arises when the knowledge graph is constructed from only English-language articles rather than from both English- and Ukrainian-language articles?

To answer these research questions, we constructed a temporal knowledge graph of supply-chain relationships centred on Ukrainian drone manufacturers identified through a review of public sources. To construct the graph, we preprocessed data from Common Crawl News (CC-News) to obtain Ukrainian- and English-language articles captured between 2017 and 2026, retaining only those containing the names of the drone manufacturers.

The preprocessed corpus is then passed through a multistep pipeline designed to extract only relationships involving the investigated drone manufacturers or their products, along with temporal information about these relationships. This focus prevents relationships unrelated to the Ukrainian drone supply chain from being introduced into the knowledge graph while allowing the suppliers and customers of the drone manufacturers to be identified.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The pipeline stages are evaluated using manually annotated evaluation sets to determine whether the pipeline is suitable for answering the research questions and to identify the limitations that must be considered when interpreting the results obtained from the final knowledge graph. Performance is reported separately for Ukrainian-source and English-source articles and for both source languages combined, allowing us to assess the pipeline's suitability for each language.

Using the constructed knowledge graph, the first research question is answered by analysing temporal developments across two-year intervals from 2018 through 2025. We examine the numbers of new and recurring entities and relationships within each interval and compare the observed patterns with documented real-world developments in the Ukrainian drone supply chain. In addition, we analyse how many manufacturers of selected drone types are identified in each interval through maker-product relationships. This analysis assesses whether the knowledge graph can correctly identify producers of types that existed before the full-scale invasion, such as reconnaissance, strike, and agricultural drones, as well as producers of types that emerged later during the invasion, such as interceptor drones.

The second research question is answered by examining the source language of the articles supporting the relationships in the knowledge graph. We determine whether each relationship was extracted from Ukrainian-language articles, English-language articles, or both and draw conclusions from this distribution.

1.2 Related Work

This section discusses approaches that, similarly to this thesis, extract supply-chain graphs from publicly available text and highlights how they differ from the approach developed here.

Köse et al. [7] present an approach that uses a large language model (LLM) to extract supply, partnership, and ownership links between companies in the semiconductor sector with high precision. Their approach uses company webpages crawled from Common Crawl and incorporates the extracted links into a temporal network. In this network, they are able to detect, for example, a decline in links between the US and China in 2020, while observing an increase in links between the EU and China. However, their approach can only extract links between companies whose web domains have been identified before relationship extraction. It is therefore unable to extract relationships involving companies that are not known beforehand. Moreover, their approach cannot extract relationships involving entities other than companies.

In the same semiconductor domain, Jin et al. [11] propose an iterative approach that queries multilingual news articles through search APIs using enriched company-name queries. From these news articles, an LLM then extracts entities, such as companies, persons, and products, and supply-chain relationships, such as production, supply, and employment relationships. Company names newly identified through entity extraction are

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

queried in subsequent iterations, after which entities and relationships are extracted again and stored in a knowledge graph. They show that their approach achieves better coverage than Bloomberg SPLC, a prominent supply-chain relationship dataset, for companies in emerging economies such as India and mainland China. However, unlike this thesis, they do not perform temporal analysis of the resulting graph.

A similar approach by Liu and Meidani [10] also uses LLMs for entity recognition and relationship extraction. They iteratively query Google News with company names and retrieve articles from different years. From these articles, they then extract entities, their descriptions, and their roles in the supply chain, such as manufacturer or distributor. Each newly identified entity is queried again to obtain a comprehensive graph. To link entities, they establish "potential" supply relationships between two entities when they are mentioned together in the same text. This is a limitation of the approach, as this heuristic is likely to extract incorrect relationships between companies. Moreover, it does not allow for granular differentiation between relationship types, such as ownership and supply relationships.

AlMahri et al. [12] propose an approach that uses LLMs to recognise and disambiguate entities such as companies and products and to extract relationships such as supply and production relationships from news articles. The extracted relationships are then represented in a knowledge graph. The resulting graph provides visibility beyond first-tier suppliers. However, like Jin et al., they do not perform temporal analysis of the knowledge graph.

Jackson et al. [13] combine LLMs and RAG to extract a multi-tier supply-chain graph. Their approach identifies multi-tier networks for the three largest electronics contract manufacturers in the world and extracts attributes such as sector and country for each company in the graph. It relies on company websites and Wikipedia pages, as well as SEC 10-K filings and earnings-call transcripts, rather than on a large-scale, general news corpus such as that used in this thesis. They also do not perform any temporal analysis.

An alternative non-LLM approach to supply-chain extraction, proposed by Wichmann et al. [8], uses deep neural networks to predict supply and ownership relationships between entities. However, it requires extensive labelling for training and operates at the sentence level, meaning that it can capture only relationships expressed within a single sentence. Kim et al. [9] build on the data labelled by Wichmann et al. and extend their experiments by fine-tuning language models such as RoBERTa-Entity, which outperform the models presented in the original paper. Moreover, they show that, in settings with limited data availability, synthetic data generation combined with human-annotated labels can improve model performance. However, their approach has the same sentence-level limitation as that of Wichmann et al.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1.3 Structure

Chapter 2 provides the background necessary to understand large-scale corpus deduplication, the inter-annotator agreement (IAA) metrics used to assess annotations, and the Damerau–Levenshtein distance.

Chapter 3 presents the CC-News repository, which is the data source for the knowledge graph, and explains how the Ukrainian drone manufacturers investigated in this thesis were identified. It also discusses how CC-News was preprocessed to form a clean, deduplicated corpus containing only texts that include a keyword match for one of the drone manufacturers' names.

Chapter 4 introduces the information retrieval and information extraction pipeline applied to the corpus created in Chapter 3. It first explains how the corpus is filtered to retain texts about the Ukrainian drone manufacturers and then describes how different relationships are extracted from those texts. Finally, it shows how the extracted relationships are used to construct the knowledge graph.

Chapter 5 shows how evaluation datasets were created and used to evaluate the different pipeline steps described in Chapter 4.

Chapter 6 presents the outputs of the different pipeline stages, the resulting knowledge graph, and the evaluation results. It also reports temporal developments within this graph and the coverage of relationships originating from English- and Ukrainian-language articles.

Chapter 7 interprets the evaluation results and assesses whether the constructed knowledge graph is suitable for answering the research questions. It then addresses the first question by comparing temporal developments observed in the graph with real-world developments and the second by investigating the contributions of the two source languages. Finally, the limitations of our approach and directions for future work are presented.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 2

Background

Before presenting the data preprocessing, pipeline design, and evaluation approach, this chapter provides additional background on large-scale corpus deduplication, the IAA metrics used to assess annotations, and the Damerau–Levenshtein distance.

2.1 Large-Scale Corpus Deduplication

The background needed to understand the large-scale corpus deduplication approach taken in this thesis is introduced in this section. It first presents Jaccard similarity before explaining what shingles are and how MinHash efficiently estimates the similarity between sets of shingles. Finally, it explains how locality-sensitive hashing (LSH) builds on MinHash to identify candidate duplicates.

2.1.1 Jaccard Similarity

The Jaccard similarity is a statistic measuring the similarity between two sets and is calculated as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

where A and B are sets.

2.1.2 Shingles

Shingles are constructed by sliding an n-token or n-character window over the text, advancing it by one token or character at each step. The string within each window is added to a set that represents the text.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

For example, a 2-token shingling algorithm, in which each token is a word, applied to the sentence

The quick brown fox jumps over the lazy dog

would produce the shingles

```
{"The quick", "quick brown", "brown fox", "fox jumps", "jumps over",
 "over the", "the lazy", "lazy dog"}
```

2.1.3 MinHash

MinHash, first introduced by Broder [15], can be used to estimate the Jaccard similarity between the shingle sets of two texts from a larger corpus without computing either the intersection or the union of the shingle sets.

To explain how MinHash works, let A and B be sets of shingles representing two texts. Let U be the set of all shingles extracted from the corpus, with $A, B \subseteq U$. Let h denote a hash function that assigns a distinct rank uniformly at random to each shingle $u_i \in U$.

Given h, the MinHash value $m_h(X)$ can be calculated for each of the sets A and B as follows.

$$m_h(X) = \min_{u_i \in X} h(u_i), \qquad X \in \{A, B\}.$$

The MinHash values for A and B are equal exactly when one of the shingles in $A \cap B$ is assigned the lowest rank among all $|A \cup B|$ shingles. Because h induces a uniformly random ordering, each shingle is equally likely to be assigned the lowest rank. Therefore, the probability that the MinHash values for A and B are equal corresponds to their Jaccard similarity:

$$\Pr[m_h(A) = m_h(B)] = \frac{|A \cap B|}{|A \cup B|} = J(A, B).$$

Based on this observation, we can use k independently chosen hash functions $h_1, \ldots, h_k$ to generate the so-called signature of a set X, defined as

$$\operatorname{sig}(X) = \left(m_{h_1}(X), \ldots, m_{h_k}(X)\right).$$

This signature can then be used to estimate the Jaccard similarity between A and B as

$$\widehat{J}(A, B) = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}[\operatorname{sig}(A)_i = \operatorname{sig}(B)_i].$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

However, calculating this estimate for the whole corpus would still require exhaustive pairwise comparisons. Therefore, the signatures are used as input to LSH.

2.1.4 Locality-Sensitive Hashing

Based on MinHash signatures, LSH probabilistically determines whether two sets are candidate duplicates. Each signature is split into b equally sized bands of r rows, with each row corresponding to one MinHash value in the signature. These parameters therefore determine the number k of hash functions, which is equal to the signature length, as

$$k = rb.$$

Each band is then hashed using an ordinary hash function. Two sets are considered candidate duplicates if at least one pair of corresponding bands is mapped to the same bucket. Assuming negligible hash collisions, the MinHash values in these bands must therefore match exactly.

For two sets with Jaccard similarity s, the probability of their classification as candidate duplicates depends on b and r and is given by [16, Sec. 3.4]

$$P(s) = 1 - (1 - s^r)^b .$$

As shown in Figure 2.1, this probability follows an S-curve. Increasing b makes the curve more permissive because additional bands provide more opportunities for a match, increasing the probability that two candidates match at lower Jaccard similarities. Increasing r makes the curve stricter because more rows per band reduce the probability that two bands match.

The Jaccard similarity threshold t for candidate selection approximates the transition point of the S-curve, above which pairs become increasingly likely to be classified as candidate duplicates. It can be used to choose r and b [16, Sec. 3.4].

$$t \approx \left(\frac{1}{b}\right)^{1/r} .$$

Through its hashing approach, LSH avoids exhaustive pairwise comparisons across all sets. To identify candidates, it only needs to find buckets containing signature bands from more than one set. However, this efficiency comes with probabilistic behaviour that can produce candidate pairs below the chosen threshold or miss pairs above it. False positives can be resolved by post-processing the candidate pairs, whereas false negatives cannot be recovered.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

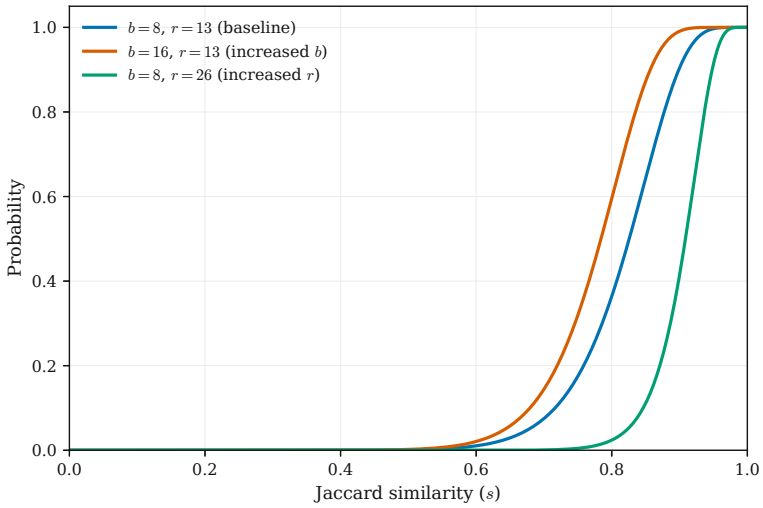

Figure 2.1: Probability of classifying two sets as candidate duplicates as a function of Jaccard similarity for three different configurations.

2.2 Inter-Annotator Agreement Metrics

The annotations in this thesis were performed by two annotators, the author and a second annotator. To assess their agreement at the document level (e.g., document relevance to a query) and the span level (e.g., named entity recognition (NER)), this thesis uses IAA metrics implemented by Prodigy, the annotation tool used in this thesis [17], [18]. The resulting scores provide an indication of the reliability and reproducibility of the annotation process, with scores above 0.8 generally considered reliable [19]. The metrics used are as follows.

2.2.1 Document-Level Agreement

Percentage Agreement

Percentage agreement measures the proportion of items on which the annotators agree. It is defined as

$$\text{Percentage agreement} = \frac{\text{Number of agreements}}{\text{Total number of items}} \times 100.$$

Krippendorff's Alpha

According to Gwet [20], Krippendorff's alpha coefficient [21] is an agreement metric that is able to handle missing data and different types of annotations, such as nominal or ordinal annotations, and can be used to evaluate agreement among more than two annotators. Fundamentally, it has the same form as Cohen's kappa, as follows.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

$$\alpha = \frac{p_a - p_e}{1 - p_e},$$

where p_a is the weighted percentage agreement between annotators, and p_e is the weighted percentage agreement that would occur by chance. Gwet [20] provides details on calculating p_a and p_e.

Gwet's AC2

Gwet's AC_2 has the same form as Krippendorff's alpha. However, it addresses some of the limitations of Krippendorff's estimation of p_e. For a detailed explanation, refer to Gwet [20].

2.2.2 Span-Level IAA

For multiclass span-level annotations, such as NER, agreement is measured using the micro-averaged F_1 score, or micro-F_1, separately for each annotated class. For this calculation, one annotator is treated as the reference and the other as the "predictor". Micro-averaging means that true-positive (TP), false-positive (FP), and false-negative (FN) counts are aggregated across all annotated documents for each class. Micro-F_1 is then calculated from precision and recall as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

2.3 Damerau–Levenshtein

The Damerau–Levenshtein distance measures how many edit operations, given a set of allowed operations, are needed to transform one string into another. The allowed operations are deletion, substitution, insertion, and transposition of two adjacent characters [22], [23]. Table 2.1 shows examples of source and target strings, the resulting Damerau–Levenshtein distances, and the operations used.

Source string	Target string	Distance	Operations applied
ring	thing	2	Insertion, substitution
thing	ring	2	Deletion, substitution
form	from	1	Transposition

Table 2.1: Examples of Damerau–Levenshtein distances and the edit operations used to transform each source string into its target string.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 3

Data Collection and Corpus Construction

This chapter introduces Common Crawl News, the main data source used in this thesis, and describes how it was preprocessed to extract the main body text from the stored HTML documents. It then presents the investigated seed nodes, i.e., Ukrainian drone manufacturers, and explains how their names and aliases were used to filter the Common Crawl corpus. Finally, it describes the approach used to deduplicate the corpus by removing exact and near-duplicate texts, producing the input corpus for the downstream pipeline.

3.1 Common Crawl News

CC-News serves as the data source for the knowledge graph pipeline. Since 2016, CC-News has collected news webpages from sources worldwide and made them available as Web ARChive (WARC) files through Amazon S3 [24], [25]. A WARC file comprises multiple records. Each record corresponds to one scraped news article and contains header fields and a payload that typically contains the news article in HTML format. In addition to the payload, the header fields `WARC-Record-ID`, `WARC-Date`, and `WARC-Target-URI` are important for our analysis. In this thesis, we refer to `WARC-Record-ID` as the article ID, `WARC-Date` as the capture date on which the article was scraped, and `WARC-Target-URI` as the article URL.

To construct a fixed input corpus, we selected all WARC files released between 1 January 2017 and 23 February 2026 at 17:16:18 GMT. In total, these files contain around 1,023,071,235 unique article IDs and have a combined size of several terabytes. This makes downloading all WARC files before processing them impossible on our infrastructure. We therefore downloaded and processed the data in small batches using DataTrove [26].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

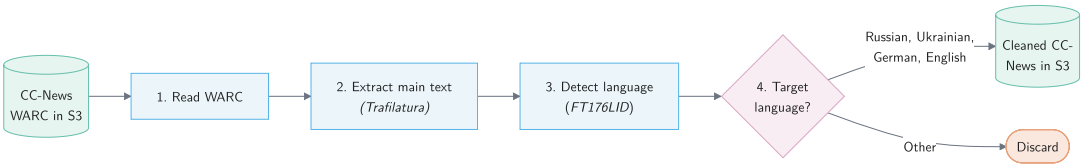

Figure 3.1: CC-News main-text extraction pipeline implemented with DataTrove.

DataTrove is an orchestration framework that separates processing into pipelines and executors. A pipeline combines modular blocks such as readers, HTML text extractors, filters, deduplication methods, custom functions, and writers. An executor runs the pipeline in parallel across the configured number of workers, either on the local machine or a Slurm cluster. The implemented preprocessing pipeline is shown in Figure 3.1.

Step 1 downloads each WARC file and passes each individual record, which consists of the HTML payload and header fields, to the subsequent pipeline stages.

Step 2 extracts the main text from the HTML using Trafilatura [27], [28]. The default parameter settings are used except for `deduplicate`, which is set to `True`, and `include_tables` and `include_comments`, which are set to `False`.

Step 3 classifies the language of each text using the fastText language classifier [29], [30]. Step 4 then retains Russian, Ukrainian, German, and English articles and writes the remaining 574,439,924 articles to S3. Ukrainian and English are directly relevant to the development of the pipeline. Russian articles may become relevant to future extensions of the pipeline, while German articles are generally relevant to the collaborating research institute ASCII.

3.2 Seed Nodes

Seed nodes are the companies and organisations for which supply-chain relationships are extracted in this thesis. As no comprehensive list or registry of all Ukrainian drone manufacturers is available, we constructed our own list through manual research using sources such as the Ukrainian Drone Ecosystem Directory [31] and the Tech Force in UA member directory [32]. We identified a total of 82 companies, which represent a subset of all Ukrainian drone manufacturers. Each of the companies was assigned a unique ID (seed node ID).

For each seed node, we identified common names and aliases in Ukrainian and English from its website or related news articles. To increase recall, we omitted common legal entity suffixes. We identified a total of 162 names and aliases, which are listed in Table A.1 in Appendix A. This list may not cover every name or alias.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.3 Seed Node Filtering and Quality Filter

To further reduce the corpus size before applying any retrieval methods, we retain only texts containing at least one seed-node name or alias. We also apply filters to improve the overall data quality. This pipeline, again implemented in DataTrove, is shown in Figure 3.2.

Step 1 removes articles likely to contain JSON-encoded content rather than plain text by excluding article URLs ending with `.json`, `.jsonl`, `.jsonld`, or `.geojson`. Step 2 retains only texts written in either Ukrainian or English.

Step 3 normalises both the seed nodes and the texts to maximise recall for Step 4. This involves normalising Unicode text to NFC form, applying case folding, removing quotation marks and apostrophe variants, replacing punctuation and symbols with spaces, converting line breaks to spaces, collapsing consecutive whitespace, and removing leading and trailing whitespace.

Step 4 performs a regex search on each normalised text for exact occurrences of normalised seed-node names or aliases. It retains texts containing at least one match and links the matched seed nodes and their seed node IDs to the article ID.

Step 5 applies a modified version of the Gopher quality filter [33] to remove texts that do not structurally resemble conventional news articles, such as those consisting primarily of bullet points. It does not assess journalistic quality of the articles. The original filter parameters were selected to extract high-quality English text for training an LLM, whereas our aim is to retain informative multilingual news articles. Because this goal requires a less strict filter, we constructed a small calibration dataset and tuned the filter parameters for our setting, as described in Appendix B.

Step 6 ensures that downstream retrieval methods operating on full texts and those operating on chunks return the same set of articles. The chunking methods described in Section 4.1.1 must locate a seed-node match in the tokenised text before creating chunks. Because some matches found in the normalised text in Step 4 cannot be reconstructed after tokenisation, the affected articles would be discarded during chunking because no chunks could be created for them. Step 6 therefore retains only articles containing a seed-node match in tokenised text, using the same tokenisation and matching strategy as the chunking methods. The matching strategy is illustrated in Algorithm C.1 in Appendix C.

After running the pipeline, 2,456,120 articles remain for deduplication. For six of the 82 seed nodes, no articles were found, or all identified articles were filtered out. These seed nodes are Justifier Drones, SKLBR, Smart UAV Technology, Svart Electronics Plus, Ukrtekhno-Atom, and vidchayduhy.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

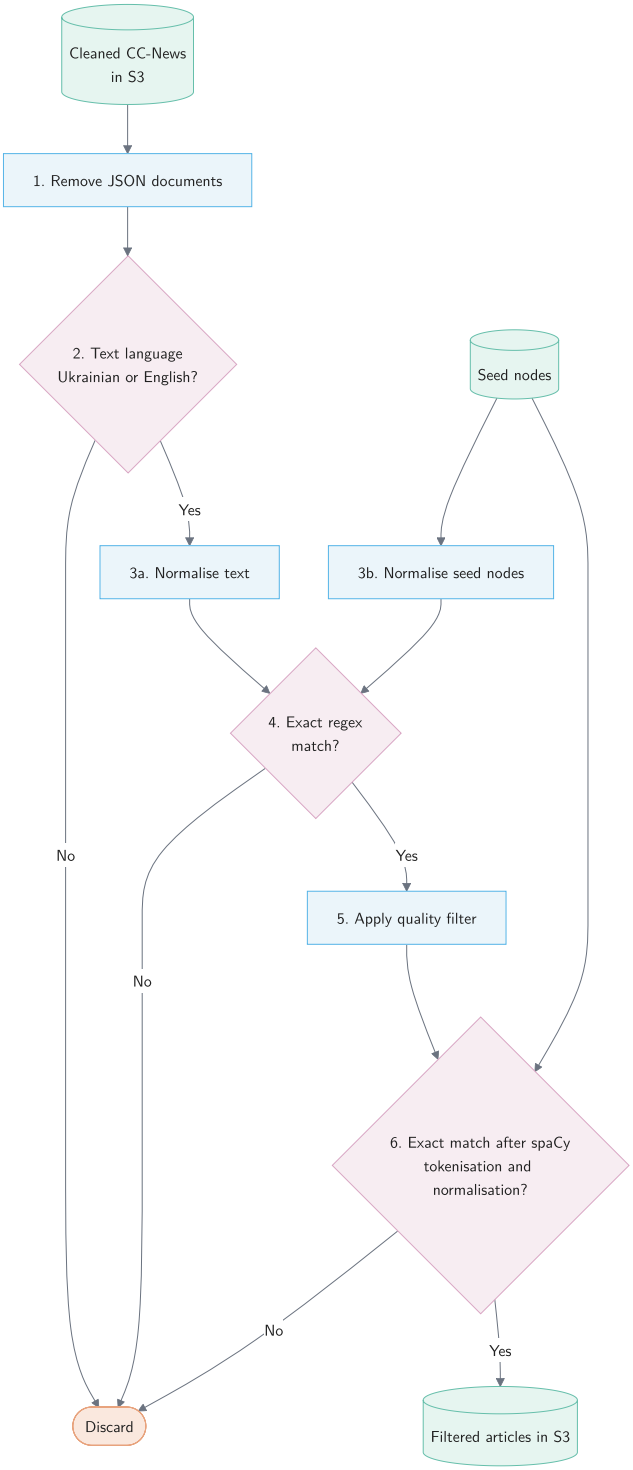

Figure 3.2: Overview of the seed-node and quality filtering pipeline.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.4 Deduplication

Inspection of the corpus during preprocessing revealed many exact and near-duplicate articles. Retaining these articles increases computational costs while providing no additional information for exact duplicates and often little additional information for near duplicates.

We define two texts as duplicates when the Jaccard similarity between their sets of normalised five-token shingles generated by DataTrove's `minhash` module is at least 0.85. However, computing the exact Jaccard similarity for every pair of the 2,456,120 articles would require $O(n^2)$ comparisons and was infeasible. Therefore, deduplication was performed in two more efficient stages. The first stage uses MinHash and LSH, which are described in Section 2.1, to form preliminary clusters of potential duplicates. The second, deterministic stage selects one representative from each cluster, ensures that every other article in the cluster has a Jaccard similarity of at least 0.85 to that representative, and retains the representative in the corpus.

MinHash and LSH were performed using DataTrove's `minhash` module. We configured LSH with eight bands and 13 rows per band, resulting in an approximate transition point at a Jaccard similarity of 0.85. After generating the candidate pairs, DataTrove transitively grouped the associated articles into preliminary duplicate clusters and left the remaining articles as singletons.

The second deduplication stage consists of four steps. First, each preliminary cluster is partitioned so that articles remain in the same cluster only if their assigned seed-node IDs are identical. This prevents articles associated with different seed nodes from being considered duplicates. The exact implementation is provided in Algorithm D.1 in Appendix D.

Next, the longest eligible article in each cluster is selected as its representative, with article length, measured as the number of tokens, used as an indicator of information content. For clusters containing fewer than 10 articles, all articles are eligible. For clusters containing at least 10 articles, a maximum length threshold u is calculated as

$$u = Q_3 + 1.5\,\mathrm{IQR}, \qquad \mathrm{IQR} = Q_3 - Q_1,$$

where Q_1 and Q_3 are the first and third quartiles of the article token counts within the cluster.

Only articles whose lengths do not exceed u are eligible. This restriction excludes overly long articles that might contain repeated text. The minimum cluster size of 10 was chosen to make the quartile estimates more robust (Algorithm D.2).

In the third step, the exact Jaccard similarity between the representative and every other article in each cluster is calculated. If the similarity is below the threshold, the article is removed from the cluster and placed in a candidate group. Each candidate group contains only articles rejected from the same cluster (Algorithm D.3).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In the final step, each candidate group is processed separately by first ordering its articles by decreasing length. The longest article is then selected as the representative of the first new cluster. Each remaining article is compared with this representative and assigned to its cluster if their similarity is at least the threshold. If its similarity is below the threshold, the article becomes the representative of another new cluster, against which the remaining unassigned articles are compared. This process continues until every article has been assigned (Algorithm D.4).

These four steps guarantee that every article in a cluster has an exact Jaccard similarity of at least 0.85 to its representative. However, representatives of different clusters could still have a Jaccard similarity of at least 0.85 because the algorithm does not compare them with one another. Consequently, some duplicates likely remain. Overall, the deduplication retained 1,619,785 articles.

3.5 Composition of the Preprocessed Corpus

Figure 3.3 provides an overview of the composition of the corpus. Figure 3.3(a) shows that a large share of the articles is associated with only four seed nodes, namely Falcons, AVATAR, Mara, and TORO. This is because they have highly ambiguous names that also match articles outside the Ukrainian drone domain. Figure 3.3(b) shows that the corpus consists predominantly of English-language articles.

The temporal distribution in Figure 3.3(c) does not indicate a sustained increase in the number of articles following the start of the full-scale invasion. Although there is a prominent peak in late 2022, it is unlikely to be related to drones. Figure F.1 in Appendix F shows that this period coincides with unusual increases in articles associated with Falcons and AVATAR. Falcons is also the name of an American football team, while the second Avatar film was released around that time.

Table 3.1 illustrates how information is structured in the corpus. Each row represents one article and exactly one associated seed node. We refer to this combination as an article–seed-node pair. Each pair is processed separately because we aim to extract information from the article that specifically relates to the associated seed node.

Article ID	Article URL	Capture date	Language	Extracted text	Seed-node ID
a-01	https://example.com	2024-03-15	en	Example text …	s-01
a-01	https://example.com	2024-03-15	en	Example text …	s-02

Table 3.1: Schematic representation of two article–seed-node pairs associated with the same article.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

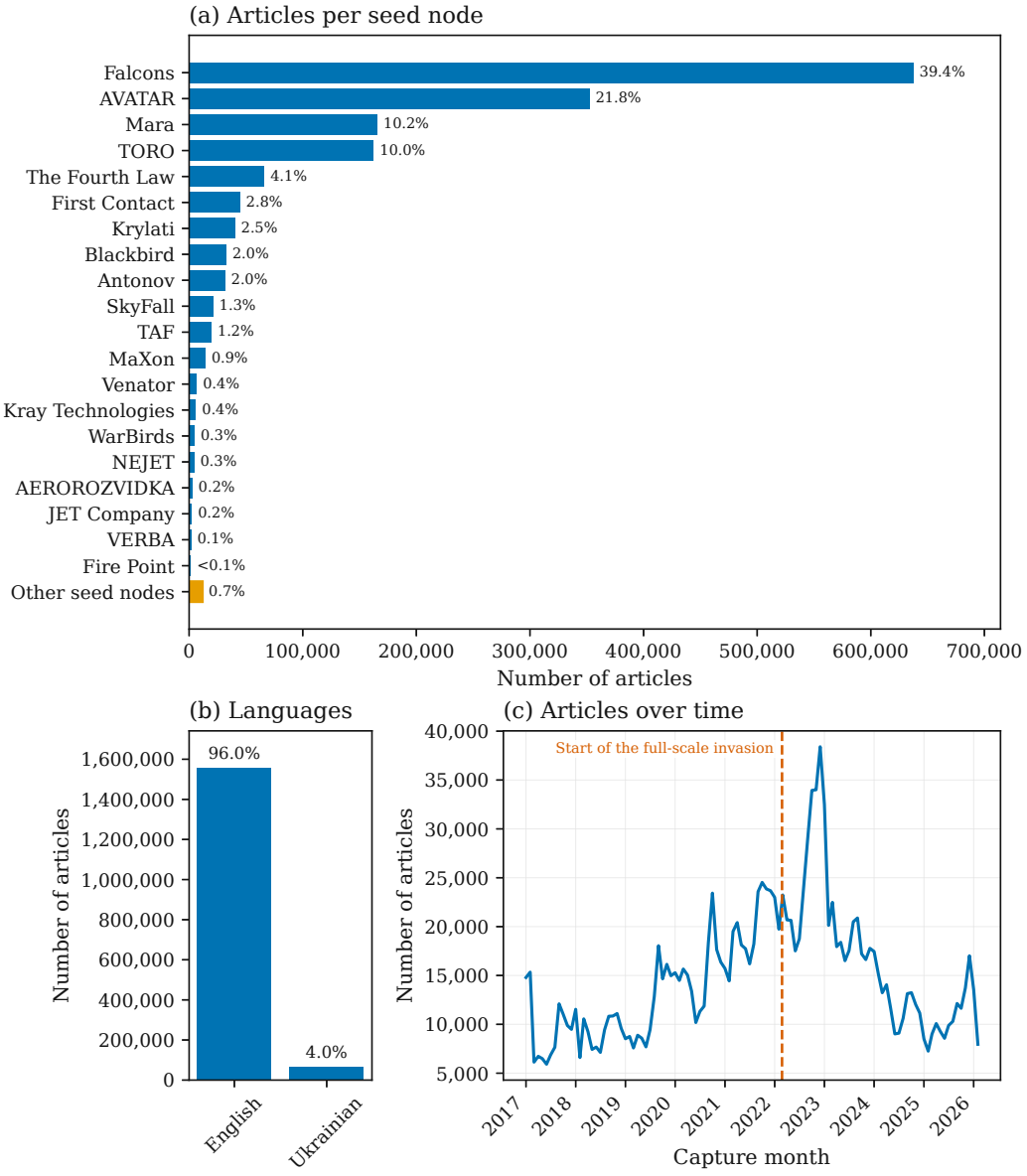

Figure 3.3: Composition of the preprocessed corpus after deduplication by (a) seed node, (b) language, and (c) capture month. Articles associated with multiple seed nodes are counted once for each seed node in (a).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 4

Knowledge-Graph Construction Pipeline

This chapter presents the pipeline and methods used to construct the knowledge graph from the preprocessed CC-News corpus. Figure 4.1 provides an overview of the pipeline, which consists of an information retrieval stage and an information extraction stage. The information retrieval stage first uses a retriever to rank article–seed-node pairs and then applies a fixed ranking cut-off, reducing the corpus size while minimising the loss of information related to the seed nodes. An LLM classifier then removes pairs in which either the matched string does not refer to the seed node or the article discusses the seed node only in the context of business activities unrelated to drones.

In the information extraction stage, an LLM removes all text passages from the articles that do not provide information about the associated seed node. The retained Ukrainian passages are translated into English. This is done because neither annotator spoke Ukrainian, and evaluating the NER and relationship extraction (RE) tasks requires the same text to be used for both prediction and annotation. After translation, the NER stage extracts and classifies entity mentions. Here, it also extracts common nouns that denote product types, which differs from the classical definition of NER. The extracted entity mentions, together with the relevant passages, then serve as input to RE, which extracts relationships that involve the seed node or one of its products. Entity mentions that are involved in relationships and refer to the same entity are then resolved. Relationships are deduplicated based on the resolved entities, after which the knowledge graph is constructed.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

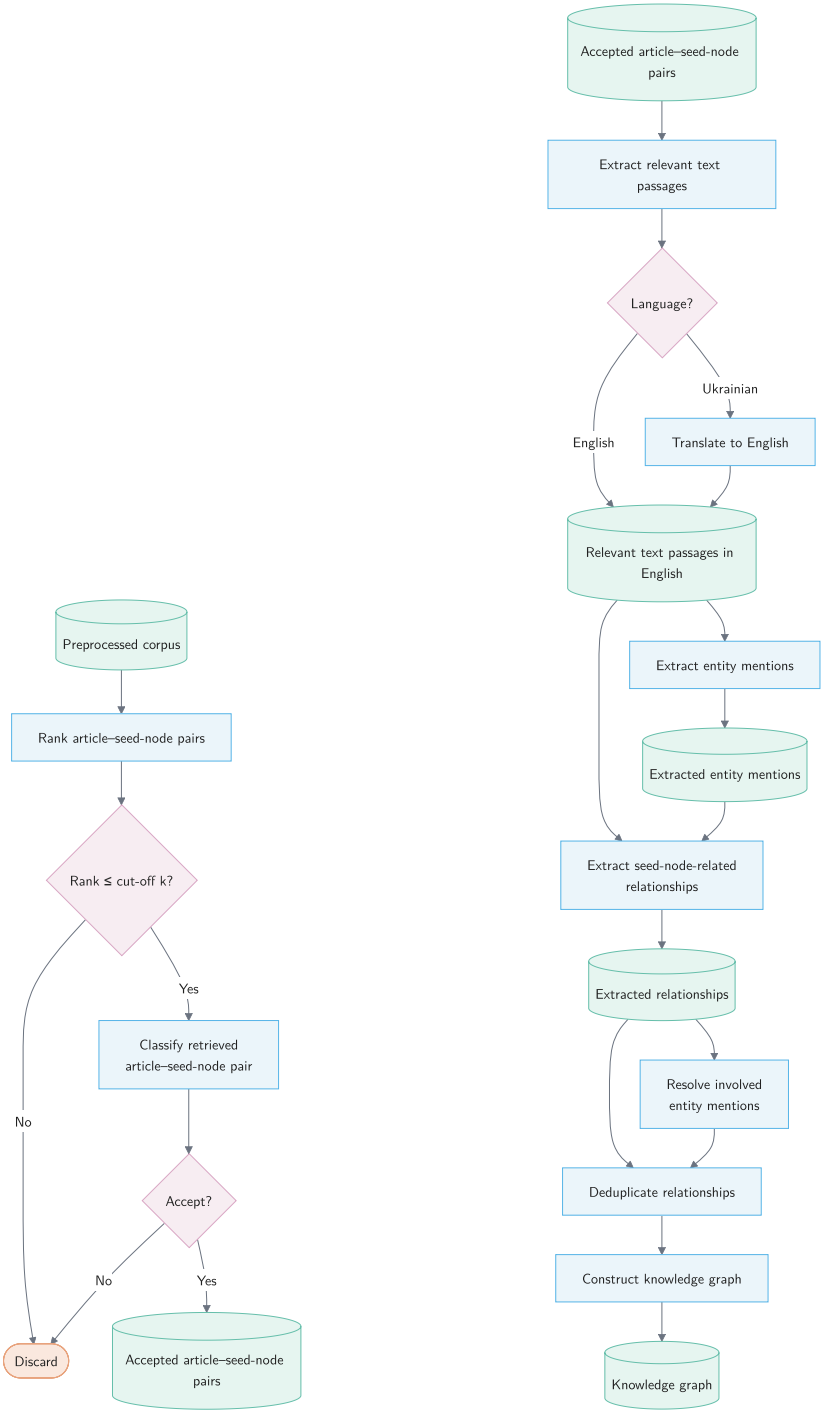

Figure 4.1: Overview of the pipeline used to construct the knowledge graph based on the preprocessed corpus.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.1 Document Retrieval

The deduplicated corpus constructed in Chapter 3 contains 1,619,785 unique articles, corresponding to 1,633,767 article–seed-node pairs, many of which do not contain information about the investigated drone manufacturers. Its size would make processing it directly with the downstream LLM-based relevance classifier too expensive. Therefore, this retrieval stage was introduced to reduce the corpus size by retrieving documents that contain information about the seed nodes.

Because an article can be associated with multiple seed nodes but contain relevant information about only some of them, the retrieval stage ranks article–seed-node pairs rather than individual articles. It produces separate rankings for English and Ukrainian articles and retains the 25,000 highest-ranked pairs from each language, resulting in 50,000 candidates for relevance classification.

The following subsections describe how texts were represented, how queries were created, how the three retrieval methods were implemented, and how their scores were aggregated into rankings. Section 5.1 in Chapter 5 describes the experiments used to compare the different retrieval configurations constructed from these queries and text representations.

4.1.1 Text Representations

We compared three text representations described in this section. These comprised full texts and chunks produced using two different methods. For both chunking methods, the maximum chunk size was 256 spaCy-generated tokens [34]. Centred-window chunking positions chunks around the seed node, whereas paragraph chunking constructs them from paragraphs. Each chunk is assigned a unique ID.

Full Text

The texts underwent no additional chunking or preprocessing and were used as they appeared in the preprocessed CC-News corpus.

Centred-Window Chunking

Unlike typical window chunking, the method does not simply split a document into overlapping windows. Instead, Algorithm C.2 in Appendix C locates all positions of a seed node in the spaCy-tokenised text. Each returned position is then used to construct a 256-token window centred on the seed node.

If an occurrence is located less than half a window's length from the beginning of the document, the window is extended to the right by the number of tokens that are "unused" on the left. Similarly, if an occurrence is located less than half a window's length from the end of the document, the window is extended to the left. If the document contains fewer than 256 tokens, no chunking is applied.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

To prevent near-duplicate chunks, a ban radius of 26 tokens, equivalent to approximately 10% of the chunk size, is applied around each seed-node position. For any other seed-node occurrence within this radius, no chunk is constructed.

Paragraph Chunking

The paragraph-chunking strategy first splits each article into paragraphs at line breaks. Each paragraph containing a seed node is then kept as an individual chunk if it does not exceed the 256-token limit. When two or more paragraphs are consecutive, every contiguous combination of these paragraphs that remains within the 256-token limit is additionally constructed.

If a paragraph exceeds 256 tokens, the largest possible window made up of complete sentences is constructed around the seed node. The initial window begins at the start of the sentence containing the token 128 positions to the left of the seed-node position and ends at the end of the sentence containing the token 128 positions to its right. If this window exceeds 256 tokens, the sentence whose outer boundary token is furthest from the seed-node position (i.e., the first token of the leftmost sentence or the last token of the rightmost sentence) is removed. This process continues until the limit is met. As in centred-window chunking, windows are extended to the right when the seed node occurs close to the beginning of the document and to the left when it occurs close to the end. If the sentence containing the seed node exceeds 256 tokens, centred-window chunking without the ban radius is used instead.

4.1.2 Query Generation

To retrieve article–seed-node pairs that contain information about the seed node, we abstracted the task into the search goal of finding texts that contain information about the Ukrainian drone industry, its ecosystem, or relationships within it. This abstraction was necessary because no information about the seed node itself was provided other than that it was a participant in the Ukrainian drone industry. We expressed this search goal in eight different queries. Each query has a keyword representation and natural-language representations based on single-query and multi-query strategies.

Keyword Queries

We constructed the eight keyword queries from eight topic-specific keyword groups. Table 4.1 provides an overview of these groups and their English and Ukrainian keywords. GPT-5.5 translated the English keywords into Ukrainian with reasoning set to medium, following the prompt in Appendix I.1.

We designed the keyword groups to provide different angles on the search goal. The *domain-specific* keyword group targets article–seed-node pairs mentioning unmanned aerial vehicles (UAVs). The *organisation* and *manufacturing* groups focus on references to companies or organisations and to manufacturing activities, respectively. The *country*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Keyword group	English keywords	Ukrainian keywords
Country	Ukraine, Ukrainian	Україна, український
Domain-specific	interceptor, reconnaissance, surveillance, strike, payload, kamikaze, bomber, loitering, drone, aerial, unmanned, FPV, UAV, UAS, quadcopter	перехоплювач, розвідка, розвідувальний, спостереження, ударний, камікадзе, бомбардувальник, баражуючий, дрон, безпілотник, БПЛА, безпілотний, FPV, ФПВ, UAV, БпАК, UAS, квадрокоптер
Organisation	company, organisation, organization, nonprofit, business, volunteer, firm, startup, enterprise, bureau, foundation	компанія, організація, неприбуткова, некомерційна, підприємство, волонтер, волонтерський, фірма, стартап, бюро, фонд, фундація
Manufacturing	producer, manufacturer, designer, developer, maker, fabricator, integrator, factory, workshop, production, assembly, engineering, industry	виробник, конструктор, розробник, виготовлювач, інтегратор, завод, фабрика, майстерня, цех, виробництво, складання, інжиніринг, інженерія, промисловість, індустрія
Supply relationship	supplier, supply, buyer, buy, purchase, acquire, customer, provides, contract, delivers, sell, seller, procurement, distributor	постачальник, постачання, покупець, купувати, закупівля, придбання, придбати, замовник, клієнт, постачає, контракт, договір, продавати, продавець, дистриб'ютор
Affiliation	owner, founder, representative, associated, affiliated, employee, employed, director, CEO, CTO, investor, shareholder, executive, management	власник, засновник, представник, пов'язаний, афілійований, працівник, співробітник, працевлаштований, директор, CEO, CEO, гендиректор, CTO, техдиректор, інвестор, акціонер, керівник, топменеджер, керівництво, менеджмент
Company location	located, headquartered, based, registered, address	розташований, штаб-квартира, базується, зареєстрований, адреса, місцезнаходження
Part of	component, part, contains, uses, module	компонент, деталь, містить, включає, використовує, модуль

Table 4.1: English and Ukrainian keyword groups used to construct the keyword queries. GPT-5.5 generated the Ukrainian keywords on 14 May 2026 with reasoning set to medium, using the system prompt in Appendix I.1 and the respective keyword-group descriptions in Table I.1.

group reflects the focus of the use case on Ukraine. The *supply relationship*, *affiliation*, *company location*, and *part-of* groups aim to rank article–seed-node pairs highly when they contain keywords indicating mentions of these relationships. We selected these keyword groups because the corresponding relationship types were planned for the final knowledge graph.[1]

To avoid combinations that were not meaningful for the use case, such as a query consisting only of terms from the *company location* group, and to limit the total number of queries, we did not combine the keyword groups using a full cross-product. Instead, the

[1]We reduced the scope of the final knowledge graph during development. We excluded affiliation and part-of relationships from the final graph, but they are intended for future inclusion.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

domain-specific and *country* groups formed a base query included in every keyword query. The *supply relationship*, *affiliation*, *company location*, and *part-of* groups formed a single relationship group. The *organisation* and *manufacturing* groups remained separate.

The power set of the *relationship*, *organisation*, and *manufacturing* groups was then constructed. Each combination was appended to the base query and concatenated into a single string, producing eight keyword queries (Q1–Q8). The resulting queries and the keyword-group combinations from which they are constructed are shown in Table 4.2.

Query	Country	Domain-specific	Organisation	Manufacturing	Relationship
Q1	✓	✓	–	–	–
Q2	✓	✓	✓	–	–
Q3	✓	✓	–	✓	–
Q4	✓	✓	✓	✓	–
Q5	✓	✓	–	–	✓
Q6	✓	✓	✓	–	✓
Q7	✓	✓	–	✓	✓
Q8	✓	✓	✓	✓	✓

Table 4.2: Keyword-group combinations defining queries Q1–Q8. The complete keyword groups are provided in Table 4.1.

Natural-Language Queries

Because an embedding model may struggle to capture the semantics of keyword groups concatenated into a single string, we generated natural-language formulations for each of the eight keyword queries (Q1–Q8) to test whether they improve retrieval performance. To do so, we implemented two generation strategies. The single-query strategy generates one natural-language representation for each keyword query. Because a single formulation may not capture all semantic concepts expressed by longer keyword queries, we also implemented a multi-query strategy that represents each keyword query using a set of four distinct natural-language queries. We selected four without empirically evaluating the effects of using a different number of queries and therefore cannot state that four is the optimal value.

The corresponding prompts, used with GPT-5.5 at the medium reasoning setting, are provided in Appendix I.2. An example English output generated from Q2 is shown in Figure 4.2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Single Natural Language Query

Ukrainian companies, nonprofits, volunteer organizations, startups, and foundations developing or operating unmanned aerial drone systems such as FPV quadcopters, UAVs, UAS, reconnaissance and surveillance drones, interceptor drones, loitering munitions, kamikaze strike drones, bomber drones, and payload-carrying aerial platforms.

Multiple Natural Language Queries

1. *A Ukrainian defense technology company develops unmanned aerial systems for reconnaissance and surveillance, including FPV quadcopters, interceptor drones, bomber payload carriers, and loitering kamikaze strike UAVs.*
2. *Profiles of Ukraine-based startups, firms, and volunteer foundations supporting battlefield drone programs, from aerial reconnaissance platforms to unmanned strike systems that deliver explosive payloads or act as one-way loitering munitions.*
3. *A nonprofit organization or business in Ukraine supplies UAS equipment for the military, combining surveillance drones, interceptor aircraft, FPV attack quadcopters, bomber drones, and kamikaze UAVs in its operations.*
4. *Reports on Ukrainian enterprises and volunteer bureaus building or procuring aerial drones for combat roles, including reconnaissance, target observation, interception, bombing payload delivery, and loitering strike missions.*

Figure 4.2: Example single-query and multi-query outputs generated for Q2 on 15 May 2026 using GPT-5.5 with reasoning set to medium. The Q2 keyword query was passed as user input to the respective prompts provided in Appendix I.2.

4.1.3 Retrieval Methods

We implemented three retrieval methods. Sparse retrieval uses BM25 [35], dense retrieval uses BGE-M3 [36], and hybrid retrieval combines the sparse and dense scores.

Sparse Retrieval

Sparse retrieval uses BM25, which was implemented with the BM25S library [37] using its default Lucene BM25 implementation and parameters. It is applied separately to each language and each of the three text representations, namely full texts, centred-window chunks, and paragraph chunks.

Before tokenisation and indexing, the corpus for each text representation is deduplicated again using exact text matching. Although the corpus was deduplicated during preprocessing, an article and its chunks may appear multiple times when the article is associated with multiple seed nodes. Removing these duplicates prevents document-frequency counts from being inflated.

The deduplicated texts are then tokenised using language-specific spaCy tokenisers. Language-specific stop words provided by spaCy are removed, as are tokens representing quotation marks, brackets, currency symbols, whitespace, URLs, email addresses, or

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

numbers. Token normalisation then differs by language.

For English, text tokens are stemmed using the Snowball stemmer [38] provided by PyStemmer [39]. We selected stemming because it reduces the number of distinct terms and matches inflected forms, such as *drone* and *drones*, while requiring less computation than lemmatisation.

For Ukrainian, PyStemmer did not provide a stemmer at the time of implementation. Text tokens are therefore lemmatised using a spaCy pipeline backed by pymorphy3 [40]. The pipeline does not include a part-of-speech (POS) tagger because the same normalisation procedure has to be applicable to keyword queries, which lack sentence context. Without POS information, pymorphy3 falls back to a lookup lemmatiser, which returns a lemma only if POS information is not needed to determine the correct form.

Following normalisation, a separate inverted index is constructed for each language and text representation. At retrieval time, each keyword query undergoes the same tokenisation, filtering, and normalisation as the indexed texts. The resulting rankings of the deduplicated text representations are then mapped back to all original article–seed-node pairs.

Dense Retrieval

All chunks produced by centred-window or paragraph chunking are embedded once with BGE-M3, a multilingual embedding model, without applying any text normalisation beforehand. We excluded full texts from embedding because of their higher computational cost and expected lower retrieval performance compared with chunk embeddings.

At retrieval time, in experiments using either a keyword query or a query produced by the single-query natural-language strategy, the query embedding is compared with each chunk embedding using cosine similarity, which defines the dense score of the chunk. For the multi-query strategy, each of the four natural-language queries is embedded separately, and the highest cosine similarity between a chunk embedding and any of the four query embeddings defines the chunk score.

Hybrid Retrieval

Hybrid retrieval combines the scores produced by sparse and dense retrieval. First, the different rankings need to be matched. The matching procedure depends on the text representations used to produce the respective rankings. When sparse and dense rankings use the same chunking method, their entries are matched by chunk ID, ensuring that scores for the same chunks are combined. When sparse retrieval uses full texts, its ranking can be matched with a dense ranking produced using either chunking method based on article IDs and seed-node IDs. Rankings based on different chunking methods are not matched.

After matching, the sparse and dense scores are min–max normalised separately to the interval $[0, 1]$. The hybrid score for each matched pair is calculated as

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$s_{\text{hybrid}} = \alpha \widetilde{s}_{\text{sparse}} + (1 - \alpha)\widetilde{s}_{\text{dense}}, \tag{4.1}$$

where $\widetilde{s}_{\text{sparse}}$ and $\widetilde{s}_{\text{dense}}$ denote the min–max-normalised sparse and dense scores, and α denotes the weight of the sparse score and can range from 0 to 1.

4.1.4 Score Aggregation and Ranking

For retrieval setups involving chunks, chunk-level scores are aggregated into article–seed-node scores by retaining the highest score. Full-text sparse scores require no aggregation and are assigned directly to the corresponding article–seed-node pairs.

The pairs are ranked separately by language in descending order of score. Ties are broken by the article ID, followed by the seed-node ID and, if chunking is used, the chunk ID.

After ranking, the 25,000 highest-ranked article–seed-node pairs per language were retained, resulting in 50,000 pairs being retrieved. This cut-off was set based on the expected cost of downstream LLM-based relevance classification.

4.2 Relevance Classification

The 50,000 article–seed-node pairs output by the retrieval stage still include pairs in which either the matched string does not refer to the seed node or the article discusses the seed node in a context unrelated to its drone-related activities. Accepting such pairs poses two risks. In the first case, relationships involving a different entity may be attributed to the seed node, resulting in incorrect relationships in the knowledge graph. In the second case, relationships and entities unrelated to the drone industry may be added to the graph, potentially distorting the analysis of how the industry developed over time.

To filter out these pairs, we introduced an LLM-based classifier using GPT-5 mini with reasoning set to medium. Each article–seed-node pair was classified once. For each pair, the model receives the full, unchunked article, the seed node's lowercased name and aliases, an entity description, and a specified business activity. A pair is kept only if both of the following criteria are satisfied.

1. The article explicitly mentions and clearly identifies the specified seed node based on the provided name or aliases, the entity description, and the surrounding context, rather than just containing a matching string or referring to a different entity.
2. The article clearly links the seed node to the specified business activity rather than discussing it only in relation to an unrelated activity.

The classifier received the same business activity for every seed node, "Production, development, manufacturing, or design of drones and unmanned aerial vehicles." It also

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

received the same entity-description template, "`{seed node}` is a Ukrainian entity (e.g., company, volunteer group, NGO, organization) that manufactures, produces, develops, or designs drones." In this template, `{seed node}` was replaced by the seed node's name. These formulations were the most specific descriptions that could be used for every seed node without researching each individually.

The model returned a JSON object containing a Boolean relevance decision and a concise reason for the classification decision. The complete classification prompt is provided in Appendix I.3.

4.3 Relevant Segment Selection

Relevant segment selection takes the article–seed-node pairs retained by the relevance classifier as input and removes the parts of each article that do not contain information about the seed node. The process first adds numbered segment tags to each article. An LLM then selects relevant segments according to relevance criteria described later in this section. Finally, the new texts are constructed using only the selected segments from English articles and the English translations of Ukrainian articles, so that all downstream processes use English texts.

4.3.1 Segment Construction

First, the text is separated into distinct, non-overlapping segments by inserting numbered tags, such as `<Segment 1>`. The positions of these tags are determined through a multi-step process. Paragraphs identified by newline characters first form preliminary segments. Because some extracted texts lack paragraph boundaries or contain exceptionally long paragraphs, some preliminary segments may become overly long. We therefore use the 99th percentile of all preliminary segment lengths as the length threshold. For each preliminary segment s exceeding this threshold, the number n_s of smaller segments into which it is divided is calculated as

$$n_s = \left\lceil \frac{t_s}{T_{99}} \right\rceil ,$$

where t_s is the number of language-specific spaCy tokens in s, and T_{99} denotes the length threshold.

The segment is then split into smaller segments by adding complete sentences until the target length, $\lfloor t_s/n_s \rfloor$, is reached. The last added sentence remains in the segment if retaining it produces a segment length closer to the target. Otherwise, the sentence is removed and becomes the first sentence of the next segment.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.3.2 Segment Relevance Criteria and Selection

The segment-selection criteria aim to retain all segments needed to establish relationships linking the seed node or one of its products to other entities. A segment is therefore considered relevant if at least one of the following criteria applies.

C1. It directly mentions the seed node, one of its researched aliases, or an alias that the article clearly establishes.

C2. It directly mentions a product that the article clearly establishes as something the seed node makes, develops, designs, sells, supplies, or otherwise offers.

C3. It indirectly refers to the seed node or to a product that the article clearly establishes as something the seed node makes, develops, designs, sells, supplies, or otherwise offers. The reference must be unambiguously resolved from the surrounding article context.

C4. It states a relationship between the seed node and another person, organisation, or place, such as a supply or ownership relationship.

C5. It states a relationship between one of the seed node's products and another product, component, system, entity, person, organisation, or place, such as a part-of or purchase relationship.

C6. It is needed to establish or disambiguate a clear relationship across multiple segments, even if it does not state the relationship directly. It is included only if it forms part of the smallest connected set of segments needed to establish the relationship.

To illustrate the criteria, Figure 4.3 presents an excerpt from a Deutsche Welle article included in the corpus [41]. It marks each segment as relevant or non-relevant and indicates which criteria apply to the relevant segments.

Segments were selected once for each article–seed-node pair kept by the relevance classifier using GPT-5 mini with reasoning set to high and the prompt in Appendix I.4. The prompt was developed from the criteria and supplemented with general concepts and examples not tied to this specific use case, both to improve performance and to allow its application to other domains. It also prioritised recall because information lost by omitting relevant segments cannot be recovered by downstream pipeline stages. The model received the complete article in its original language with numbered segment tags, together with the seed node's name and aliases. It returned a JSON object containing the selected segment IDs and brief inclusion and exclusion reasons.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

<Segment 1> **NOT RELEVANT**

Ukraine: Huge increase in drone production

<Segment 2> **RELEVANT (C1, C2, C3)**

Work never stops in this Skyeton production facility. The private armaments manufacturer's employees work throughout the night, cutting components, assembling them, checking software. The end product is the Raybird-3 drone system, military designation ACS-3. The unmanned mini-aircraft has a wingspan of just under three meters and weighs less than half a hundredweight. It can fly up to 120 kilometers, meaning it can also be deployed for reconnaissance flights over enemy territory.

<Segment 3> **NOT RELEVANT**

Boom in the Ukrainian drone industry

<Segment 4> **RELEVANT (C1, C2, C4, C5)**

The Raybird-3 has been in use with the Ukrainian armed forces since 2018. However, since February 2022, Skyeton has significantly increased production. "Things that used to take us a year before 2022 we can now get done in a few weeks," says Andrii Fialkovskyi, CEO of Skyeton.

<Segment 5> **RELEVANT (C1, C2, C3)**

Until 2014, the firm manufactured training aircraft. Now, it produces three aircraft, a catapult start system, two ground stations, an antenna unit and a set of spare parts per week. The company's biggest challenge is maintaining security against Russian attacks, prompting Skyeton to distribute its production across various different locations.

Figure 4.3: Application of the segment-relevance criteria to an excerpt from a news article published by Deutsche Welle [41] concerning the seed node Skyeton. Light-green segments are considered relevant, while grey segments are considered non-relevant. The criterion labels indicate which criteria apply.

4.3.3 Ukrainian-to-English Translation

In parallel with segment selection, all articles in Ukrainian article–seed-node pairs were translated into English so that downstream NER and RE received only English inputs. Despite this translation, this thesis continues to distinguish between "English-source" and "Ukrainian-source" texts to indicate whether the original article was written in English or Ukrainian.

Each full article, including all segment markers, was translated to preserve the full context and keep translation independent of the segment-selection method. We used GPT-5.5 with reasoning set to high to minimise translation errors. The prompt, provided in Appendix I.8, required the segment markers to remain unchanged.

4.3.4 Segment Filtering and Text Reconstruction

After translation and segment selection, all segments not selected by the LLM are removed. To produce a continuous text, consecutive selected segments from the same

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

original paragraph are joined, while existing paragraph boundaries are preserved. Non-consecutive segments are separated by `[..]` to indicate omitted content. Finally, the numbered segment tags are removed.

4.4 Named Entity Recognition

NER follows relevant segment selection to extract entity mentions of different types, producing the second input to RE alongside the segment-filtered text.

The entity types to be extracted were selected to enable the extraction of the relationships defined in Section 4.5. Table 4.3 provides an overview of the entity types used in this thesis, as well as their definitions, boundaries, and examples. It summarises the full NER prompt provided in Appendix I.9, which contains the complete definitions, rules, and examples used for NER.

Using this prompt, NER is performed once for each article–seed-node pair using GPT-5 mini with reasoning set to high. The model receives only the text reconstructed from the selected segments and returns the predictions as a JSON array. Each array element contains the predicted entity type, the text span, and a reason for its selection. Because the output does not include positional information, repeated occurrences of the same span assigned to the same entity type cannot be distinguished. The model is therefore instructed to return such occurrences only once. Each returned array element is then converted into an entity-mention tuple of the form `(entity_type, span)` and linked to the corresponding article–seed-node pair.

4.5 Relationship Extraction

In the second-to-last pipeline step, an LLM uses the entity mentions identified during NER to extract four types of supply-chain relationships and form alias groups containing mentions that refer to the same underlying entity. The relationship types form a minimal knowledge graph schema that captures which organisations produce which products, which organisations receive these products, the product types to which named products belong, and where organisations are located. Table 4.4 presents the relationship definitions, directions, source and target entity types, attributes, and examples. The complete definitions, examples, and additional rules are provided in the prompt in Appendix I.10.

As shown in Table 4.4, the relationships include attributes. The `date_mention` attribute enables temporal analysis by recording a `DATE` mention indicating when the relationship existed. For `supplies_to` relationships, the supplied product is also extracted, enabling more detailed analysis of the business relationship between the organisations.

Only relationships involving the seed node or one of its products are extracted, preventing unrelated relationships between the other entity mentions passed to the model from being included in the knowledge graph. Because Ukrainian-source texts were translated into English, mentions of the seed node might no longer appear in their original form.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Therefore, alongside the seed node names and aliases, the model is provided with a Boolean flag indicating whether the text was translated, allowing it to infer how the seed node is mentioned in the translated text.

The step also extracts alias groups for potential downstream use in entity resolution. For each entity type, all entity mentions referring to the same underlying entity are grouped together. Alias groups are returned only if at least one of their entity mentions occurs in an extracted relationship. Additional rules for forming alias groups are provided as part of the RE prompt in Appendix I.10.

To extract the relationships and alias groups, each article–seed-node pair is processed once using the previously mentioned prompt and GPT-5 mini with reasoning set to high. The model receives the text reconstructed from the pipeline-selected segments, the lowercased seed node name and aliases, a flag indicating whether the text was translated, and the entity mentions predicted by the preceding NER step, which are grouped by entity type.

The model returns a JSON object with separate arrays for relationships and alias groups. Each relationship is represented as an object containing its type, source, target, and any attributes supported by the text, as well as a reason for the prediction. Relationships with the same type, source, and target are returned once, together with all supported attributes found across the text. Alias groups are returned as objects, each containing the entity type and a list of spans predicted as aliases.

Each returned relationship is then converted into a dictionary in which every source, target, and attribute entity mention is represented using the `(entity_type, span)` tuple introduced for NER. For example, a `makes` relationship may be represented as follows.

```
{
    "relationship": "makes",
    "source": ("ORG", "DeViro"),
    "target": ("NAMED_PRODUCT", "Leleka-100"),
    "attributes": [
        ("date_mention", ("DATE", "2015"))
    ]
}
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Entity type	Definition	Boundaries	Examples
ORG	Named companies, institutions, organisations, and other structured groups. Common nouns referring to these groups are also included when made specific by a proper noun or proper adjective.	Geopolitical entities, including when they act like an organisation in the text.	**Athlon Avia**, **NATO**, and **Ukrainian army** are ORG mentions, whereas "Ukraine" in "Ukraine announced . . . " is not.
NAMED_PRODUCT	Specifically named products, models, software, or systems.	–	The span **Furia** in "the Furia reconnaissance drone" is a NAMED_PRODUCT mention.
PRODUCT_TYPE	Categories of concrete products, components, systems, and software, including modifiers that define their function or subtype.	Abstract categories and capabilities are not considered product types. Modifiers describing quality, origin, ownership, or condition are not included in the annotation.	The span **reconnaissance drone** in "the Furia reconnaissance drone" is a PRODUCT_TYPE mention. In "cheap drone", only **drone** is a PRODUCT_TYPE mention. In "strike drone", the full span **strike drone** is a PRODUCT_TYPE mention. **All-wheel drive system** is a PRODUCT_TYPE mention, whereas "all-wheel drive" alone is not because it denotes only a capability. Broad terms such as "payload" are not PRODUCT_TYPE mentions.
DATE	Dates, month–year combinations, and standalone years containing an explicit year. Date ranges are annotated as separate dates.	Dates without a year and other time expressions.	**01.01.2001**, **April 2010**, and **2026** are DATE mentions, whereas "Monday", "June", and "10 April" are not. In "2022–2026", **2022** and **2026** are separate DATE mentions.
GPE	Countries, cities, states, provinces, municipalities, and other political or administrative regions.	Place-derived adjectives and non-administrative geographical or cultural regions and places.	**Vienna** and **Ukraine** are GPE mentions, whereas "Ukrainian", "the Alps", and "Western Europe" are not.

Table 4.3: Entity definitions and examples used in the NER pipeline step.

Relationship type	Definition	Source type	Target type	Attributes	Example	Example extraction

`makes`	An organisation creates, designs, develops, manufactures, assembles, or otherwise produces a named product or product type.	`ORG`	`NAMED_PRODUCT` or `PRODUCT_TYPE`	The `date_mention` attribute records any `DATE` mentions clearly linked to the relationship.	"DeViro already made the Leleka-100 in 2015."	`source: DeViro`
						`target: Leleka-100`
						`date_mention: 2015`

`supplies_to`	One organisation directly supplies, sells, delivers, or otherwise provides a named product or product type to another organisation.	`ORG`	`ORG`	The `named_product` and `product_type` attributes record supplied `NAMED_PRODUCT` and `PRODUCT_TYPE` mentions, respectively. At least one must be present. The `date_mention` attribute records any `DATE` mentions clearly linked to the relationship.	"TFL delivers 1,000 TFL-1 autonomous guidance systems to Vyriy."	`source: TFL`
						`target: Vyriy`
						`named_product: TFL-1`
						`product_type: autonomous guidance systems`

`is_type`	A named product is an instance of or belongs to a product type.	`NAMED_PRODUCT`	`PRODUCT_TYPE`	–	"TFL delivers 1,000 TFL-1 autonomous guidance systems to Vyriy."	`source: TFL-1` `target: autonomous guidance systems`
`located_in`	An organisation has its headquarters, registered office, branch, factory, or another physical facility in a geopolitical entity.	`ORG`	`GPE`	The `date_mention` attribute records any `DATE` mentions clearly linked to the relationship.	"TFL is based in Kyiv."	`source: TFL` `target: Kyiv`

Table 4.4: Relationship definitions and their attributes, source entity types, and target entity types used for RE.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.6 Knowledge Graph Construction

The final pipeline step constructs the knowledge graph from the extracted relationships. However, because different entity mentions across these relationships may refer to the same underlying entity, they must first be resolved. The following sections therefore first describe the entity-resolution procedure and then explain how its output is used to construct the knowledge graph.

4.6.1 Lexical Entity Resolution

For entity resolution, we initially planned to use the alias groups extracted in the RE stage. However, the model's performance in predicting aliases was considered insufficient, and the predicted alias groups were not used (see also Section 6.5). We therefore grouped entity mentions using a conservative lexical matching approach.

Before lexical matching, all entity mentions occurring in the extracted relationships are collected and deduplicated. Each mention is then normalised by applying Unicode NFC normalisation, removing invisible characters, canonicalising variants of dashes, quotation marks, slashes, and ellipses, and converting the mention to lowercase. Whitespace is added before and after separators such as slashes, consecutive whitespace is collapsed, and hyphens are replaced with whitespace. Quotation marks at the beginning or end of a word and parentheses are removed. Legal suffixes are additionally removed from normalised `ORG` mentions using the Cleanco package [42].

The entity mentions are then grouped by entity type, building candidate pools. Within each pool, they are sorted first by the length of their normalised representation, then by that representation, and finally by the original entity mention. A greedy procedure is applied separately to each pool. In each iteration, the next mention in the pool is selected as the representative of a new entity group. The remaining mentions serve as candidates and are compared with the representative using the lexical matching procedure described below. Matching candidates are assigned to the group and removed from the pool. Once all candidates have been compared, the group is complete, and the next iteration begins. This process continues until the pool is empty.

To determine whether a representative and candidate match, the numerical sequences extracted from the two mentions must first be identical in both order and value. This prevents, for example, products with different version numbers from matching. If this check passes, the two mentions are split into tokens at whitespace. If they contain different numbers of tokens, the candidate is rejected. Otherwise, their tokens are compared pairwise in their original order. Each token pair must satisfy at least one criterion listed below. If any token pair fails to satisfy a criterion, the candidate is rejected.

1. **Exact match.** The candidate token is identical to the representative token.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2. **Approximate match.** The Damerau–Levenshtein distance (see Section 2.3) between the candidate and representative tokens, calculated using RapidFuzz [43], does not exceed the threshold determined by the length of the representative token. The threshold is
   - 0 if the representative token has up to 5 characters (only an exact match is possible),
   - 1 if it has 6–10 characters, and
   - 2 if it has more than 10 characters.
3. **Singular–plural match.** For `PRODUCT_TYPE` mentions, the final token pair also matches if *inflect* [44] identifies a singular–plural relationship.

The Damerau–Levenshtein distance thresholds were design choices and were not empirically validated. Because the threshold depends on the length of the representative token, matching is not always symmetric, making a deterministic order crucial. Table 4.5 illustrates this asymmetry alongside other examples of matches and non-matches.

Representative	Candidate	Match	Non-match explanation
deep strike drone *deep strike drone*	deep-strike drones *deep strike drone*	✓	–
AQ 400 Scythe *aq 400 scythe*	AQ-400 Scythe *aq 400 scythe*	✓	–
Leleka *leleka*	Lelek *lelek*	✓	–
Lelek *lelek*	Leleka *leleka*	×	The representative token `lelek` contains only five characters, so an exact match is required.
An-32B *an 32b*	An-32P *an 32p*	×	The representative token `32b` contains fewer than six characters, so an exact match is required, but the corresponding candidate token is `32p`.
light attack drone *light attack drone*	night attack drone *night attack drone*	×	The representative token `light` contains fewer than six characters, so an exact match is required, but the corresponding candidate token is `night`.
57th motorized brigade *57th motorized brigade*	58th motorized brigade *58th motorized brigade*	×	The numeric sequences differ (`57` and `58`).

Table 4.5: Examples of representative–candidate pairs, whether they are matched by the entity-resolution algorithm, and explanations for non-matches. The italicised text shows the normalised form. Tokens are formed by splitting the normalised form at whitespace.

After all groups have been created, each group is treated as a single entity and assigned an ID. To select the name displayed for the entity in the knowledge graph, the original,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

non-normalised entity mention occurring most frequently across all extracted relationships is used. Each original, non-normalised entity mention is then linked to the ID and display name of the corresponding entity.

4.6.2 Graph Materialisation in Neo4j

The knowledge graph was materialised in Neo4j, which was deployed in a Docker container and populated using the Neo4j Python driver [45], [46]. Using these entity-resolution results, each entity mention in an extracted relationship was replaced with the ID of its resolved entity. The relationships were then deduplicated based on these entity IDs, and the attributes of duplicate relationships were merged.

After deduplication, the resolved `ORG`, `NAMED_PRODUCT`, `PRODUCT_TYPE`, and `GPE` entities were materialised as Neo4j nodes. The `is_type`, `located_in`, and `makes` relationships were represented as directed pairwise relationships between these entity nodes.

A `supplies_to` relationship connects multiple entities, namely one supplier `ORG`, one recipient `ORG`, and at least one supplied `PRODUCT_TYPE` or `NAMED_PRODUCT`. It therefore cannot be represented as an ordinary pairwise relationship. For each combination of supplier, recipient, and product, an intermediate `SUPPLY` node was created and connected to the three corresponding entity nodes, as shown in Figure 4.4.

`DATE` entity mentions were not materialised as nodes. Instead, years extracted from them were stored in the `year_mentions` property of the corresponding relationships and the entity nodes involved in them. When RE did not identify a `date_mention` for a relationship, the capture date of the article was stored in the `article_dates_without_year_mentions` property of the entities and relationships. Both properties are later used for the temporal analysis of relationships and entities.

Alongside temporal information, details such as the number of supporting articles and their texts were stored for each relationship or entity in the additional properties listed in Table 4.6.

Property	Description
`document_evidence_count`	Number of distinct articles supporting the entity or relationship.
`evidence_texts_json`	JSON-encoded records of supporting articles, including the article ID, seed node ID, language, URL, relevant text, and capture date.
`year_mentions`	Unique years extracted from a relationship's `date_mention` attributes.
`article_dates_without_year_mentions`	Capture dates of supporting articles when no `date_mention` was extracted for the entity or relationship.

Table 4.6: Selected properties relevant to knowledge-graph analysis.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

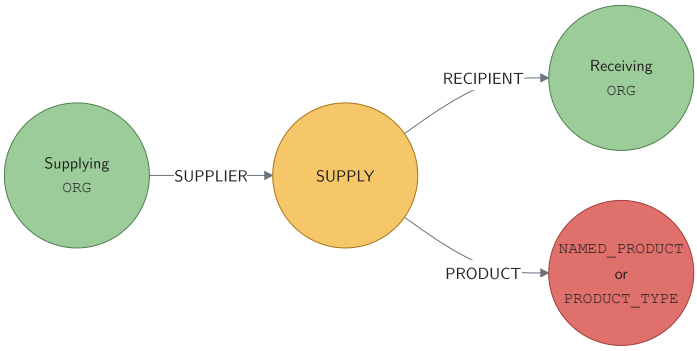

Figure 4.4: Representation of a supply relationship using an intermediate `SUPPLY` node.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 5

Pipeline Evaluation Design

This chapter describes how the pipeline steps discussed in the preceding chapter were evaluated to assess the suitability of our methods for constructing the supply-chain knowledge graph. It first details which retriever configurations were evaluated, how the LLM-labelled dataset used for retriever selection and evaluation was constructed and validated, and how this dataset was used to select and evaluate the final retriever configurations. For each subsequent pipeline stage, it first describes how the corresponding evaluation dataset was constructed and then explains how it was used to evaluate that stage. The evaluation results are then presented in Chapter 6.

5.1 Document Retrieval

This section describes how different retriever configurations for the document-retrieval stage described in Section 4.1 were evaluated and how the final configurations were selected. It first presents the configurations chosen for evaluation and the evaluation metric, as both are needed to understand how the LLM-labelled evaluation dataset for retriever selection and evaluation was subsequently constructed. Finally, it presents the procedures used to select the final retriever configurations and evaluate them on the held-out test split.

5.1.1 Evaluated Retrieval Configurations

Sparse Retrieval

For sparse retrieval, 24 configurations per language were evaluated (see Table 5.1). Each of the eight keyword queries (Q1–Q8, see also Table 4.2) was evaluated using full texts, paragraph chunks, and centred-window chunks. Only keyword queries were used because BM25 relies on lexical term matching and does not consider the semantic meaning of the query. Natural-language queries were therefore not expected to improve retrieval

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and would additionally have introduced nondeterminism through their generation. As BM25 can process full texts, the full-text representation was included to evaluate whether sparse retrieval benefits from chunking in our setting.

Component	Evaluated options	Number of options
Query	Q1–Q8	8
Query representation	keyword	1
Text representation	full text, paragraph chunks, or centred-window chunks	3
	Total configurations per language	**24**

Table 5.1: Overview of sparse retrieval configurations.

Dense Retrieval

Dense retrieval was evaluated using 48 configurations per language, as shown in Table 5.2. In addition to keyword queries, dense retrieval used single-query and multi-query natural-language representations of the query, which aim to convey the semantic intent of the query more effectively than keyword queries. Both natural-language query representations were generated ten times to assess variation in retrieval performance resulting from the probabilistic nature of LLM-based query generation. The full-text representation was excluded because of its computational cost and expected lower performance than the chunked representations.

Component	Evaluated options	Number of options
Query	Q1–Q8	8
Query representation	keyword, single-query, or multi-query	3
Text representation	paragraph chunks or centred-window chunks	2
	Total configurations per language	**48**

Table 5.2: Overview of dense retrieval configurations.

Hybrid Retrieval

Hybrid retrieval had the largest configuration space because it combined the results of both the sparse and dense retrieval configurations, producing 6,912 configurations per language (see Table 5.3). All combinations of sparse and dense queries and query representations were evaluated. Paragraph chunks were paired only with paragraph chunks, and centred-window chunks only with centred-window chunks, while full unchunucked article sparse

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

retrieval scores were paired with both dense chunking methods. The sparse-score weight α from Equation 4.1 was evaluated at values $\{0.1, 0.2, \ldots, 0.9\}$.

Component	Evaluated options		Number of options
	Sparse	Dense	
Query	Q1–Q8	Q1–Q8	64
Query representation	keyword	keyword	3
	keyword	single-query	
	keyword	multi-query	
Text representation	full text	paragraph chunks	4
	full text	centred-window chunks	
	paragraph chunks	paragraph chunks	
	centred-window chunks	centred-window chunks	
Sparse-score weight α	$0.1, 0.2, \ldots, 0.9$		9
	Total configurations per language		**6,912**

Table 5.3: Overview of hybrid retrieval configurations. Sparse and dense options shown on the same row are paired.

5.1.2 Retrieval Evaluation Metric

To reflect the objective of retrieving as many relevant article–seed-node pairs as possible, we evaluated retrieval performance using recall. However, labelling the entire corpus was infeasible. We therefore calculated recall using the relevant article–seed-node pairs in our evaluation dataset, whose construction is described in Section 5.1.3, and treated all unlabelled pairs as non-relevant. Consequently, we can compare only how well the retrievers recover the annotated relevant pairs and cannot draw conclusions about their performance across all relevant article–seed-node pairs. Nevertheless, the results provide an indication of their relative retrieval performance.

As retriever performance is usually reported at a given cut-off k and natural-language query generation was repeated ten times, potentially leading to performance differences across runs, performance was reported using $\overline{\text{Recall@}k}$, which denotes Recall@$k$ averaged across runs of the same configuration. The measure is defined as

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

$$\overline{\text{Recall@}k}(c) = \frac{1}{n_c} \sum_{i=1}^{n_c} \text{Recall@}k(c, i),$$

where c denotes an evaluated configuration, i identifies one of its retrieval runs, and n_c is 10 for configurations using natural-language queries because query generation was repeated ten times, and 1 for keyword-query configurations because their queries were fixed. For keyword-query configurations, $\overline{\text{Recall@}k}$ is therefore equivalent to ordinary Recall@k.

5.1.3 Construction of the Retrieval Evaluation Dataset

This section explains how we constructed SilverFinal from two datasets created in different ways to provide broad coverage of relevant article–seed-node pairs because no labelled data were available for this use case. The first dataset, SilverBase, used random sampling stratified by seed node, while the second, SilverExtension, used retrieval-based sampling to broaden coverage and reduce the bias introduced by SilverBase.

We labelled both silver sets using LLM-based relevance judgements, which made it feasible to annotate more than 75,000 samples. This choice is supported by Upadhyay et al., whose findings indicate the potential for LLMs to replace fully manual TREC-style assessments when estimating run-level retrieval performance [47]. To assess the reliability of the LLM judgements, we validated them using human annotations, as described in Section 5.1.4.

SilverBase

In the corpus, seed nodes with less ambiguous names, such as WeTrueGun and MRIYA 4.5.0, are associated with fewer articles than seed nodes with more ambiguous names, such as AVATAR (Figure F.2). Based on this observation, we constructed SilverBase by sampling up to the same number of articles for each seed node within a language. This means that, for seed nodes associated with fewer articles than the maximum sample size per seed node, all associated articles were annotated by the LLM. Assuming that the LLM provided sufficiently accurate judgements, this allowed all relevant articles for these seed nodes to be identified. For seed nodes associated with more articles than the maximum sample size, random sampling still gave relevant articles a chance of being included in the annotations.

Because the English corpus contains substantially more articles than the Ukrainian corpus, we sampled up to 2,000 English and 200 Ukrainian articles per seed node. For both languages, all associated articles were annotated for approximately 80% of the seed nodes.

GPT-5 mini generated the relevance judgements for the sampled articles using the LLM-based procedure described in Section 4.2 and the prompt provided in Appendix I.3, with

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the full, unchunked article as input. To limit API costs, we ran the model only once per article despite the stochasticity of its outputs.

After labelling, we split SilverBase by article ID into an 80% development set and a 20% test set. Across its development and test splits, SilverBase contains 45,202 article–seed-node pairs. Among them, 3,701 (8.2%) are labelled as relevant, including 2,339 Ukrainian and 1,362 English pairs.

Sampling SilverBase independently of retrieval rankings avoided introducing bias towards particular retrievers. However, the fixed sample sizes meant that all articles or a larger share of them were labelled for seed nodes with fewer associated articles, while only a subset was labelled for seed nodes with more associated articles. This may underrepresent relevant articles for these seed nodes and introduce bias towards seed nodes with fewer associated articles.

SilverExtension

To reduce the bias in SilverBase, we constructed SilverExtension by sampling and labelling article–seed-node pairs from retrieval rankings using a pooling-like approach. Its purpose was to identify additional relevant pairs for seed nodes underrepresented in SilverBase.

The approach constructs a candidate pool for LLM-based annotation by sampling highly ranked article–seed-node pairs from multiple retrieval rankings. Combining candidates from multiple rankings reduces bias in retriever selection because a pool constructed using only one method would favour that method, as unannotated pairs potentially retrieved by other methods are treated as non-relevant during our evaluation.

To capture a broad range of potentially relevant pairs, it is beneficial to sample from rankings that place different article–seed-node pairs near the top. We expected configurations using different queries to produce this diversity because the queries express different angles on the search goal. We therefore reduced the number of configurations considered for the pooling-like approach by grouping sparse and dense retrieval configurations separately by language and query (Q1–Q8), while hybrid retrieval configurations were grouped by language and query combination.

To select the best-performing configuration from each group, we first applied each configuration to the corpus to produce a ranking and then evaluated it on the relevant article–seed-node pairs in the SilverBase development split using $\overline{\text{Recall@10,000}}$. The best-performing configuration in each group was retained. This yielded eight sparse and eight dense configurations per language, one per query, and 64 hybrid configurations per language, one per query combination.

To produce samples for annotation, we applied the sampling procedure described in Algorithm E.1 in Appendix E separately to each language. First, within each retrieval method, the selected configurations were ordered by decreasing $\overline{\text{Recall@10,000}}$. Starting with the best-performing configuration, the top 10,000 results from each ranking were

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

iteratively added to a candidate pool until it exceeded 20,000 unique article–seed-node pairs. For each pair x, the number of processed rankings that included it among their top 10,000 results was recorded. Based on this occurrence count, its sampling probability was calculated as

$$p(x) = \frac{c(x)}{\sum_{y \in \mathcal{U}} c(y)},$$

where $\mathcal{U}$ denotes the candidate pool and $c(x)$ denotes the occurrence count of pair x.

From each candidate pool, 10,000 pairs were then sampled according to the assigned probabilities to limit the number of required annotations. This meant that pairs retrieved by multiple configurations were more likely to be included in the final sample, while pairs retrieved by only one or a few configurations could still be selected, increasing sample diversity. After sampling, the samples drawn for each method were combined and deduplicated, and pairs already present in SilverBase were removed to form SilverExtension.

The ranking cut-off of 10,000, the minimum pool size of 20,000 unique pairs, and the sample size of 10,000 pairs per retrieval method were design choices and were not empirically validated. The chosen ranking cut-off was intended to limit the inclusion of non-relevant pairs at lower ranks. Stopping once the pool exceeded 20,000 unique pairs limited its size and prevented lower-performing configurations from contributing candidates. The sample size of 10,000 pairs per method represented a trade-off between annotation costs and coverage. Note that other values for the ranking cut-off, pool size, and sample size could also have produced a valid SilverExtension.

After sampling, SilverExtension was labelled using the same procedure as SilverBase and was also split into a development set and a test set in the same way. In total, SilverExtension comprises 30,592 article–seed-node pairs. Its 1,212 relevant pairs (4.0%) consist of 189 English and 1,023 Ukrainian pairs.

SilverFinal

SilverFinal was created by combining the corresponding development and test splits of SilverBase and SilverExtension. No article–seed-node pair can appear in both the development and test sets, although the same article can occur in both when associated with different seed nodes.

SilverFinal contains 75,794 article–seed-node pairs, divided into 60,626 development and 15,168 test pairs. English pairs account for 72.2% of the dataset. Of the 4,913 pairs (6.5%) labelled as relevant, 3,362 are Ukrainian and 1,551 are English pairs. Of these relevant pairs, 3,937 belong to the development split and 976 to the test split.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.4 Human Validation of LLM-Based Relevance Judgements

We validated the LLM-based relevance judgements in SilverFinal by manually labelling a subset of 1,000 article–seed-node pairs. Since only approximately 6.5% of the pairs were labelled as relevant, simple random sampling would have directed most of the annotation effort towards non-relevant pairs. It would also have included more English than Ukrainian pairs because SilverFinal predominantly consists of English pairs, although we considered both languages equally important to the use case. We therefore drew a random sample without replacement across the development and test splits containing 250 LLM-assigned relevant and 250 non-relevant pairs from each language.

Before manual annotation, Ukrainian texts were translated into English using GPT-5.5 with the reasoning parameter set to high and the prompt provided in Appendix I.6. We then manually labelled all texts in Prodigy according to the relevance criteria defined in Section 4.2. The annotation interface displayed the seed node as metadata and highlighted possible seed-node mentions identified by GPT-5.5 with reasoning set to high. For translated Ukrainian texts, the metadata also showed an explanation by the model of why the span was highlighted, as a seed node match in Ukrainian could initially be either a generic string match or an actual reference to the drone company. This is potentially confusing in the translation. For example, *Krylati* is a seed node but can also occur as part of the Ukrainian term for "cruise missile," which would be highlighted if it were used with that meaning in the original text. The respective prompts used to identify mentions for highlighting in English-source and Ukrainian-source texts are provided in Appendix I.5.

Each annotator labelled 600 article–seed-node pairs, including an overlapping subset of 200 pairs that both annotators independently labelled. Since the task used clear relevance criteria and binary document-level labels, we considered this overlap sufficient both to estimate IAA using percentage agreement, Krippendorff's α, and Gwet's AC_2, which are described in more detail in Section 2.2, and to produce a high-quality validation dataset. After annotation, the annotators resolved disagreements in the overlapping subset through discussion to create the final reference dataset.

To evaluate the LLM-based judgements, we used weighted precision, recall, and F_1 to account for the earlier balancing and adjust the metrics to reflect the original distribution in SilverFinal. To do so, each pair in the manually labelled reference dataset was assigned a weight indicating how many pairs in SilverFinal it represents for its combination of language and LLM-assigned label. For language l and LLM-assigned relevance label r, the weight was calculated as

$$w_{l,r} = \frac{N_{l,r}}{n_{l,r}} = \frac{N_{l,r}}{250}, \qquad r \in \{\text{pos}, \text{neg}\},$$

where $r = \text{pos}$ and $r = \text{neg}$ denote positive and negative LLM-assigned relevance labels, respectively. $N_{l,r}$ denotes the number of pairs in SilverFinal associated with language

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

l and label r, and $n_{l,r} = 250$ denotes the corresponding number of manually annotated pairs. Table 5.4 shows the resulting sampling weights.

Language	$w_{l,\text{pos}}$	$w_{l,\text{neg}}$
English	6.20	212.61
Ukrainian	13.45	70.91

Table 5.4: Sampling weights by language and SilverFinal label.

Based on these weights, the weighted metrics for a given language l were calculated as follows.

$$\text{Precision}_l = \frac{w_{l,\text{pos}}\,\text{TP}_l}{w_{l,\text{pos}}\,\text{TP}_l + w_{l,\text{pos}}\,\text{FP}_l},$$
$$\text{Recall}_l = \frac{w_{l,\text{pos}}\,\text{TP}_l}{w_{l,\text{pos}}\,\text{TP}_l + w_{l,\text{neg}}\,\text{FN}_l},$$
$$F_{1,l} = 2 \times \frac{\text{Precision}_l \times \text{Recall}_l}{\text{Precision}_l + \text{Recall}_l}.$$

Here, TP_l, FP_l, and FN_l denote the numbers of samples in language l classified as true positives, false positives, and false negatives, respectively.

5.1.5 Configuration Selection and Held-Out Evaluation

To select the final retriever configuration for each language, all retrieval configurations were applied separately to the English and Ukrainian portions of the full corpus. Each configuration was evaluated using the relevant pairs in the development split of SilverFinal to calculate $\overline{\text{Recall@25,000}}$. This metric represents retrieval performance at the cut-off applied to the rankings in the pipeline (see Section 4.1.4). For each language, the best-performing retrieval configuration was selected. If two or more configurations achieved the same score, k was reduced in increments of 5,000 until a single configuration achieved the highest $\overline{\text{Recall@}k}$. The selected configuration for each language was used to retrieve the article–seed-node pairs passed to the downstream pipeline.

We then evaluated the selected configuration for each language without further modification on the test split of SilverFinal using $\overline{\text{Recall@25,000}}$.

5.2 Relevance Classification

This section first describes the evaluation dataset and then details the experiments performed to evaluate the relevance classifier presented in Section 4.2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2.1 Evaluation Dataset

The relevance classifier was evaluated using the same 1,000 article–seed-node pairs annotated for the validation of SilverFinal, as described in Section 5.1.4. The final human labels served as the reference, and the relevant label was treated as the positive class.

5.2.2 Evaluation Inputs and Metrics

The classifier was evaluated in ten independent runs over the complete evaluation dataset. The model received the seed node names and aliases, the article in its original language, and the business activity and entity description specified in Section 4.2. Predictions made on Ukrainian texts were matched against the annotations of their English translations. In each run, precision, recall, and F_1 were calculated for English-source pairs, Ukrainian-source pairs, and all article–seed-node pairs combined. Unlike in the validation of SilverFinal, the metrics were calculated without weights. The mean and standard deviation across the ten runs were reported for each metric.

5.3 Relevant Segment Selection

This section first describes the annotation setup and procedure used to create the evaluation dataset for the relevant segment selection step presented in Section 4.3. It then explains how this pipeline step was evaluated.

5.3.1 Evaluation Dataset and Annotation

To create the evaluation dataset for relevant segment selection, 50 Ukrainian and 50 English article–seed-node pairs were randomly sampled from the human-annotated dataset used to validate the LLM-based relevance judgements (see Section 5.1.4).

Before annotation, segment markers were added to the texts using the procedure described in Section 4.3.1. Following this, the Ukrainian texts were translated into English using GPT-5.5 with reasoning set to high and the prompt in Appendix I.8.

The two annotators each labelled all 100 pairs in Prodigy. The interface displayed the seed node and its manually researched aliases as metadata and highlighted potential seed-node mentions using the same procedure described in Section 5.1.4. The annotators labelled segments by selecting their markers according to the criteria defined in Section 4.3.2 and the additional concepts and examples provided in the prompt in Appendix I.4. Before alignment, IAA was measured using micro-F_1, as described in Section 2.2.2. The annotations were then aligned through discussion to establish the final reference labels. In total, 486 segments were labelled as relevant.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.3.2 Evaluation Inputs and Metrics

Relevant segment selection was evaluated in ten independent runs on the evaluation dataset. In each run, the model received the seed node, its aliases, and the text with segment markers in its original language. For Ukrainian texts, predictions were evaluated against the labels assigned to the English translations. A predicted segment was considered correct if it matched a relevant reference segment for the same article–seed-node pair. Micro-precision, micro-recall, and micro-F_1 were calculated separately for English-source pairs, Ukrainian-source pairs, and all article–seed-node pairs combined in each run. The mean and standard deviation across the ten runs were reported for each metric.

5.4 Named Entity Recognition

As in the preceding sections, this section first describes in detail how the evaluation dataset for the NER pipeline step presented in Section 4.4 was created using human annotations. It then explains how NER was evaluated.

5.4.1 Evaluation Dataset and Annotation

The NER evaluation dataset was built on the evaluation dataset for relevant segment selection. To remain consistent with the pipeline, the pipeline-generated English translations were used for Ukrainian-source texts (see Section 4.3.3), and the reference annotations were used to identify the relevant segments in these translations.

The annotators then labelled each segment separately in Prodigy, using the entity types and definitions in Table 4.3 and the detailed rules and examples in the prompt in Appendix I.9 as the basis for their decisions. Annotating the segments separately reduced the cognitive burden compared with identifying all entity mentions in a complete article at once. To provide context when needed, the complete article from which each segment originated was also displayed as non-annotatable text.

IAA was calculated separately for each entity type using micro-F_1, with an entity mention counted as a true positive only if its span, position, and entity type matched between the two annotators. Following this, the annotators resolved disagreements through discussion to establish the final reference annotations. These segment-level annotations were transformed into `(entity_type, span)` tuples that were unique at the article–seed-node pair level. Repeated occurrences of the same entity type and span within an article–seed-node pair were therefore counted once.

5.4.2 Evaluation Inputs and Metrics

NER was evaluated using two input variants, with the text for each article–seed-node pair constructed either from human-selected segments or from pipeline-selected segments. The human-selected input was used to measure NER performance independently of model-based segment selection, while the pipeline-selected input was used to assess

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

how segment selection affected downstream NER performance. For both input variants, the text was constructed from the selected segments using the procedure described in Section 4.3.4.

Ten independent runs were performed for each input using the same model and prompt as in the pipeline. Each run with the pipeline-selected input used the segments predicted as relevant in the corresponding run of the relevant segment selection evaluation. The human-selected input remained unchanged across all ten runs.

A prediction counted as a true positive when its `(entity_type, span)` tuple matched a reference tuple for the same article–seed-node pair. Micro-precision, micro-recall, and micro-F_1 were calculated per entity type and overall for English-source pairs, Ukrainian-source pairs, and all article–seed-node pairs combined. Each metric was reported for every run and as the mean and standard deviation across the ten runs.

5.5 Relationship Extraction

The relationship extraction step extracts both relationships and aliases, as described in Section 4.5. This section therefore describes how the corresponding evaluation datasets were annotated, how the aliases were used to normalise relationships for evaluation, and finally how the relationship and alias predictions were assessed.

5.5.1 Evaluation Dataset and Annotation

The RE evaluation used the human-annotated entity mentions from the NER evaluation and the texts constructed from the human-selected segments. For each article–seed-node pair, the two annotators independently annotated the defined supply-chain relationships between the already annotated entity mentions using a modified Prodigy interface, which also supported the annotation of relationship attributes (see Figure 5.1). Their decisions were based on the relationship definitions in Table 4.4 and the prompt in Appendix I.10, which provides the complete rules and examples.

Based on the alias rules defined in the same prompt, the annotators additionally linked entity mentions referring to the same underlying entity using `is_alias` relationships, under the restriction that at least one of the mentions had to be involved in an annotated supply-chain relationship. This was done for two reasons. First, it provided a reference dataset for evaluating the model's alias-extraction performance. Second, it allowed relationships that differed in their selected entity mentions but referred to the same underlying entities to be matched. To enable this comparison, the alias annotations had to be aligned before the relationship annotations were compared, as described in the following subsections.

Alias Agreement and Alignment

Before the aliases were aligned, IAA was measured using micro-F_1. For this calculation, the pairwise `is_alias` relationships were first transitively grouped separately for each

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

annotator and article–seed-node pair because an entity could be mentioned in several different forms within an article. An alias group counted as a true positive when both annotators produced exactly the same group for the same article–seed-node pair.

The alias annotations were then aligned by adjusting the pairwise `is_alias` relationships through discussion using Prodigy. The agreed relationships were transitively grouped using the same procedure as before. The resulting groups served as the reference dataset for the later evaluation of the aliases predicted by the model and were also used to construct a mapping between each entity mention and the entity to which it referred within the article–seed-node pair. To construct this mapping, each entity mention in an alias group was assigned the same ID representing the underlying entity, while every other entity mention involved in a relationship was assigned its own ID.

Relationship Agreement and Alignment

To evaluate both the annotator agreement and the model-predicted supply-chain relationships, we distinguished between core and full relationships. A core relationship contains only the relationship type, source, and target, whereas a full relationship also includes its attributes. This distinction allowed us to assess how including attribute extraction in the task affected annotator agreement and model performance.

To measure the IAA, each entity mention in the annotated relationships was replaced with its corresponding entity ID from the alias mapping. The IAA was then measured separately for core and full relationships using micro-F_1.

After calculating the IAA, the annotators resolved all disagreements using a custom-written alignment tool because Prodigy cannot align relationships with attributes. The tool classified each disagreement as a source, target, or attribute disagreement or as a missing relationship. It allowed the annotators to resolve these disagreements by selecting already annotated entity mentions for the sources, targets, and attributes of relationships from drop-down menus. Figure 5.2 shows the interface. After relationship alignment, alias groups not involved in any final relationship were removed from the alias mapping.

Finally, the normalised annotations were converted into core and full reference tuples. Core tuples contained the relationship type and the source and target entity IDs. Full tuples additionally contained a sorted list of entity IDs for each attribute.

5.5.2 Evaluation Inputs and Metrics

As in the NER evaluation, two input variants were used. The human-selected input combined manually selected segments with human-annotated entity mentions to measure RE performance independently of the performance of the preceding segment-selection and NER steps. The pipeline-generated input combined pipeline-selected segments with the corresponding NER predictions to measure the overall performance of the information extraction pipeline at its final extraction step.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Using the same model and prompt described in Section 4.5, RE was performed ten times on both input variants. The human-selected input remained fixed across the ten runs, while each pipeline-generated run used the corresponding outputs from the preceding pipeline steps. In both cases, the model received the respective text and entity mentions together with the seed node, its aliases, and a Boolean flag indicating whether the text had been translated. The model returned both relationship and alias predictions, which were evaluated as described in the following subsections.

Relationship Prediction Evaluation

To compare the predicted relationships with the reference annotations, any prediction containing an entity mention that had not been supplied to the model was counted as a false positive and excluded from further comparison. Each entity mention in the remaining predictions was then replaced with its entity ID using the reference alias mapping, making the predictions directly comparable with the normalised reference relationships.

The normalised predictions were then converted into the same core and full tuple representations as the reference annotations. For each representation, a predicted tuple counted as a true positive when an identical reference tuple existed for the same article–seed-node pair.

For each run, micro-precision, micro-recall, and micro-F_1 were computed separately for English-source pairs, Ukrainian-source pairs, and all article–seed-node pairs combined, both by relationship type and overall. The mean and standard deviation across all ten runs were also reported for each metric.

Alias Prediction Evaluation

Alias predictions were evaluated directly against the reference alias groups because the model already returned each prediction as a group. A prediction counted as a true positive only when an identical reference group existed for the same article–seed-node pair. This meant that any missing or additional entity mention within a group prevented a match.

Alias performance was measured using micro-precision, micro-recall, and micro-F_1, with the same language breakdown and run-level reporting as for the relationships.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

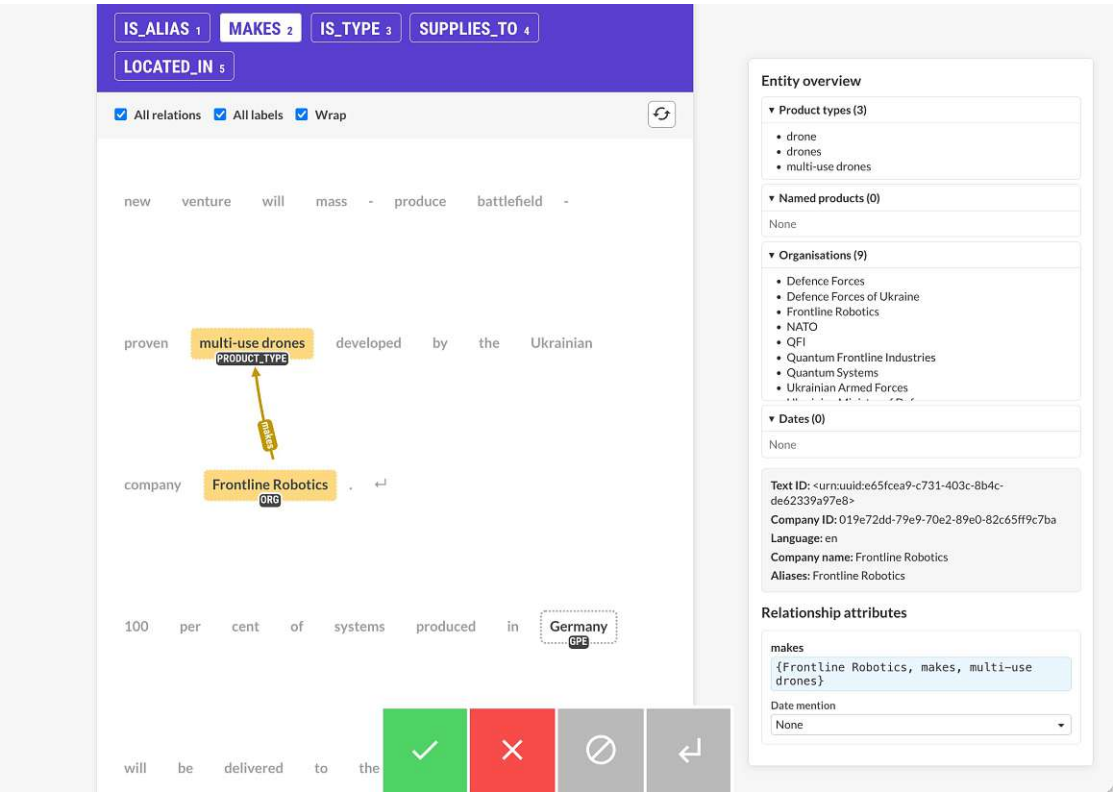

Figure 5.1: Modified Prodigy interface displaying linkable entity mentions and relationship-attribute annotation inputs for an excerpt from a Tech.eu news article [48].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

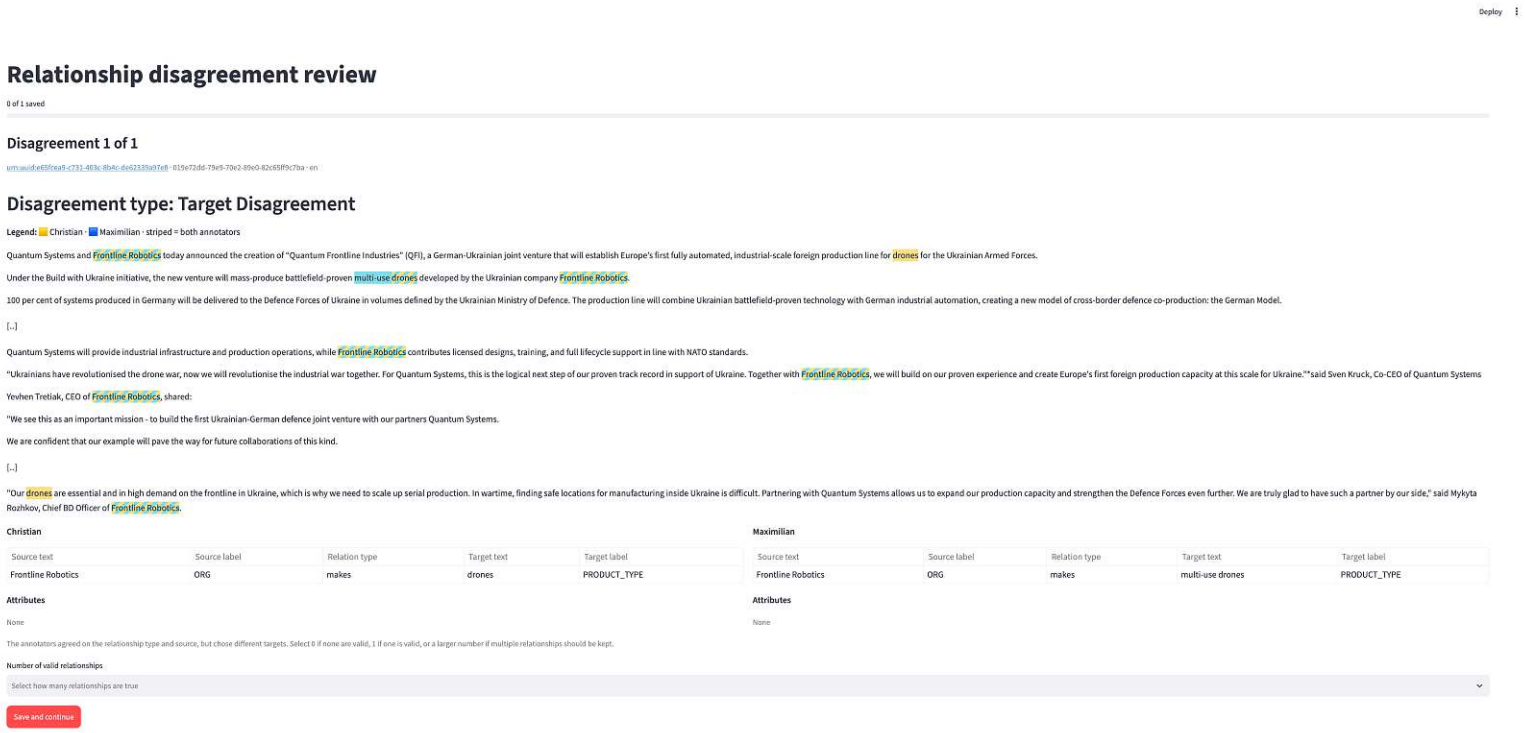

Figure 5.2: Interface of the custom relationship-disagreement alignment tool displaying excerpts from a Tech.eu news article [48].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 6

Results

This chapter first presents the evaluation results and describes the outputs of each stage of the pipeline used to construct the supply-chain knowledge graph. It then describes the resulting graph and examines its temporal development through changes in entity and relationship counts and in the number of producers identified for selected drone types. Finally, it reports relationship coverage by source language.

6.1 Document Retrieval

This section presents the IAA achieved during annotation and the composition of the human-labelled reference dataset used to validate SilverFinal, the LLM-annotated dataset used for retriever selection and evaluation. It then describes which retriever performed best for each language on the development split of SilverFinal and reports the performance of the final configurations on the held-out test split.

6.1.1 Validation of SilverFinal

Table 6.1 reports the IAA for the overlapping subset of the SilverFinal validation dataset described in Section 5.1.4.

Metric	Value
Percentage agreement	93.5%
Krippendorff's α	0.8699
Gwet's AC_2	0.8702

Table 6.1: IAA metrics for the annotations used to validate the SilverFinal dataset.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

All three metrics exceeded 0.8, indicating high agreement between the annotators. After resolving the disagreements, we labelled 477 article–seed-node pairs as relevant, of which 241 (50.5%) were Ukrainian, and 523 as non-relevant, of which 259 (49.5%) were Ukrainian.

Table 6.2 reports the weighted evaluation metrics separately for each language. All metrics strongly indicate that the model's relevance judgements align with the human relevance judgements, with the model achieving perfect recall for English. Additionally, Figure 6.1 shows the disagreements within the overlapping subset between the two annotators and between each annotator and the model. In both comparisons, the model agrees with each annotator at least as often as the annotators agree with each other. These results support using SilverFinal to estimate retriever performance.

Language	Weighted precision	Weighted recall	Weighted F_1
English	0.9360	1.0000	0.9669
Ukrainian	0.9680	0.9787	0.9733

Table 6.2: Weighted evaluation metrics for the relevance labels in SilverFinal by language.

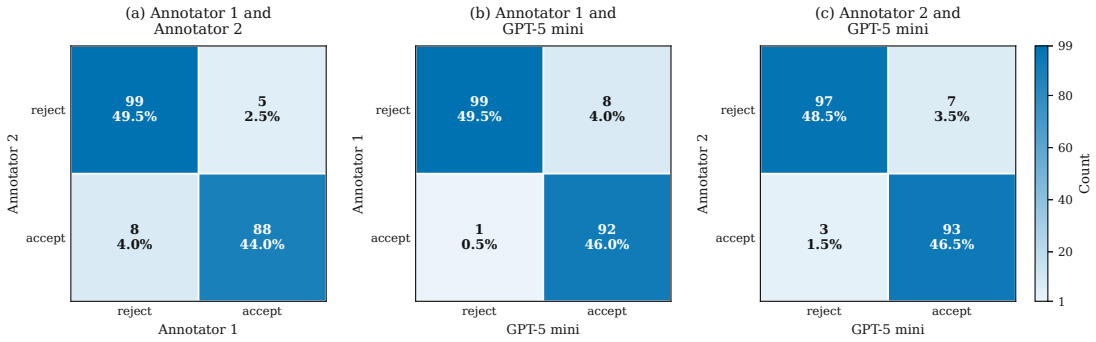

Figure 6.1: Pairwise confusion matrices for the relevance judgements on the 200 overlapping article–seed-node pairs. Matrix (a) compares the two annotators. Matrices (b) and (c) compare GPT-5 mini with each annotator.

6.1.2 Retriever Selection

Figure 6.2 compares the best-performing configuration for each method and language across cut-offs on the development split of SilverFinal, using the evaluation measure defined in Section 5.1.2. Hybrid retrieval achieved higher recall than BM25 at lower cut-offs, but their performance converged at the budgeted cut-off of 25,000 article–seed-node pairs per language, while dense retrieval remained below both.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

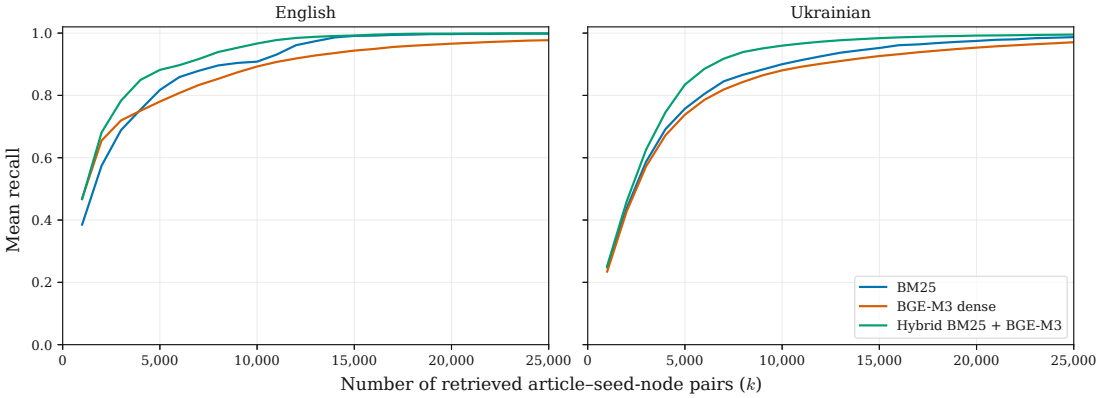

Figure 6.2: $\overline{\text{Recall@}k}$ across retrieval cut-offs on the development split of SilverFinal for the best-performing configuration of each method and language.

Table 6.3 presents the final configurations, selected from 69 English and two Ukrainian configurations tied at $k = 25{,}000$ by comparing progressively lower cut-offs as described in Section 5.1.5.

Configuration dimension	English	Ukrainian
Method	hybrid retrieval	hybrid retrieval
BM25 text representation	window chunks	full text
BGE-M3 text representation	window chunks	paragraph chunks
BM25 query	Q1	Q4
BGE-M3 query	Q3	Q4
BGE-M3 query representation	keyword	keyword
Sparse-score weight α	0.6	0.6

Table 6.3: Selected retriever configurations for English and Ukrainian.

Looking beyond the selected configurations, $\overline{\text{Recall@25,000}}$ exceeded 0.99 for 1,241 hybrid and three BM25 configurations in English and for 400 hybrid configurations in Ukrainian. The high-performing hybrid configurations used various combinations of chunking methods, queries and sparse-score weights. This indicates that their performance was not limited to a specific combination of settings and that hybrid retrieval was robust across the evaluated configurations.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

6.1.3 Retriever Evaluation

The held-out evaluation described in Section 5.1.5 showed that the selected English and Ukrainian configurations each missed only one positive article–seed-node pair at $k = 25{,}000$, achieving $\overline{\text{Recall@25,000}}$ values of 0.9969 and 0.9985, respectively, although these values do not measure full-corpus recall. Nevertheless, since SilverFinal was constructed to capture a wide range of relevant article–seed-node pairs, the high recall provides an indication that the selected retrievers surface a considerable number of relevant pairs while still missing some of them.

Because both configurations used deterministic keyword queries for their dense components, the reported $\overline{\text{Recall@25,000}}$ equals Recall@25,000, and the results do not depend on query generation.

6.2 Relevance Classification

This section first presents the evaluation results for the relevance classifier and then reports how many article–seed-node pairs were classified as relevant. Finally, it describes the composition of the resulting corpus of relevant articles.

6.2.1 Evaluation

Table 6.4 reports relevance-classifier performance by language and overall on the 1,000 manually annotated article–seed-node pairs used to validate SilverFinal, following the evaluation procedure described in Section 5.2.

Source language	Precision	Recall	F_1
English	0.9574 ± 0.0050	0.9898 ± 0.0030	0.9733 ± 0.0027
Ukrainian	0.9569 ± 0.0026	0.9859 ± 0.0035	0.9712 ± 0.0023
Combined	0.9571 ± 0.0028	0.9878 ± 0.0028	0.9722 ± 0.0019

Table 6.4: Performance of the relevance classifier over ten runs by language and across both languages. Values are reported as mean ± sample standard deviation.

As with the validation of SilverFinal, the relevance classifier achieved exceptionally high precision, recall and F_1, with little variation across the ten runs. This provides further strong evidence that its judgements closely align with human judgements, and that LLMs are well suited to this kind of task.

6.2.2 Pipeline Relevance Classification Results

After applying the classifier to the retrieval output, 4,913 article–seed-node pairs, corresponding to 4,340 unique articles, were classified as relevant. These pairs covered 68 of the initial 82 seed nodes. The average API cost across all 4,913 pairs was USD 0.0007 per pair.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Figure 6.3 shows the composition of the articles classified as relevant by seed node, language, and capture month. This composition differs substantially from that of the initial preprocessed corpus shown in Figure 3.3. Whereas English articles accounted for 96.0% of the initial preprocessed corpus, Ukrainian articles accounted for 68.5% of the articles classified as relevant. This shift in language distribution shows that restricting the analysis to English would have excluded most of the articles classified as relevant.

Only six of the 20 seed nodes that had the most associated articles before retrieval were also among the top 20 after relevance classification. In particular, the ambiguous seed nodes “Falcons”, “AVATAR”, “Mara”, and “TORO”, which dominated the initial preprocessed corpus, accounted for just 1.38%, compared with 80.94% previously. This significant change in prevalence and the considerably less skewed distribution suggest that the retrieval and relevance classification processes removed many of the unrelated articles associated with ambiguous seed-node names.

The classified articles also show a sustained increase following the start of the full-scale invasion, whereas no such trend was visible prior to it. This temporal shift is consistent with the reported growth of Ukraine’s drone sector, which expanded from seven manufacturers at the start of the invasion to over 500 by 2026 [49].

6.3 Relevant Segment Selection

This section begins by reporting the IAA and composition of the reference dataset and the corresponding evaluation results for the relevant segment selection step. Following this, it presents and interprets the pipeline output, focusing on the resulting reduction in token count.

6.3.1 Evaluation

The evaluation dataset was constructed according to the procedure described in Section 5.3. The two annotators achieved an IAA of 0.93, measured using micro-F_1. Of the 3,235 segments, 486 (15.02%) were annotated as relevant, of which 277 were from Ukrainian article–seed-node pairs.

Table 6.5 reports micro-precision, micro-recall, and micro-F_1 scores on the evaluation dataset. As expected, recall exceeded precision, reflecting the emphasis placed on recall during the development of the method, as described in Section 4.3.2. The high recall and acceptable precision suggest that the relevant segment selection process retains most segments related to the seed node while limiting the number of unrelated segments passed to the next stage of the pipeline.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

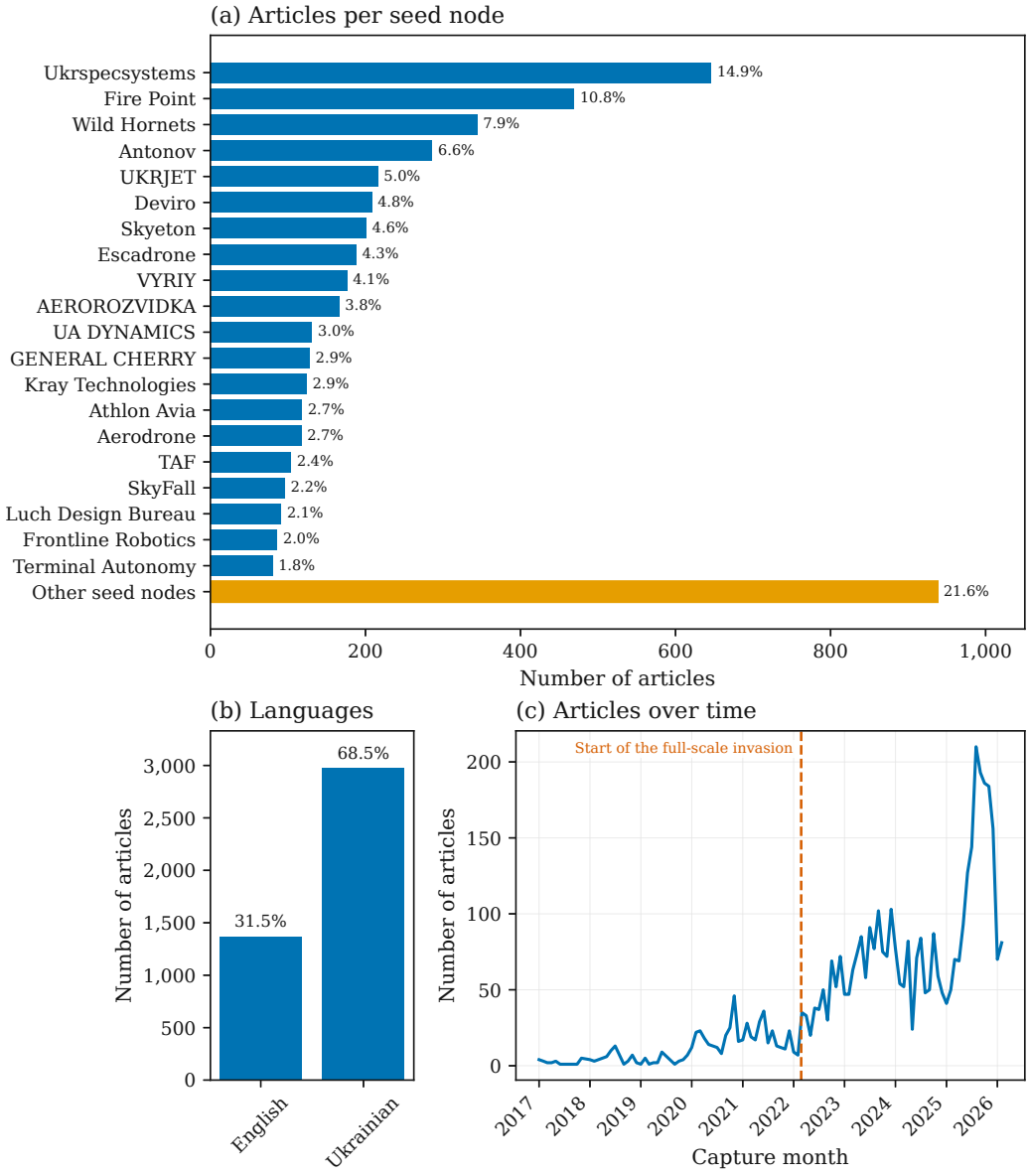

Figure 6.3: Composition of the 4,340 distinct articles with at least one positive relevance judgement by (a) seed node, (b) language, and (c) capture month. Articles classified as relevant for multiple seed nodes are counted once for each seed node in (a).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Source language	Micro-precision	Micro-recall	Micro-F_1
English	0.798 ± 0.021	0.955 ± 0.006	0.869 ± 0.012
Ukrainian	0.771 ± 0.037	0.950 ± 0.014	0.851 ± 0.026
Combined	0.782 ± 0.025	0.952 ± 0.009	0.859 ± 0.017

Table 6.5: Performance of relevant segment selection over ten runs by language and across both languages. Values are reported as mean ± sample standard deviation.

6.3.2 Pipeline Relevant Segment Selection Results

Table 6.6 reports the total counts of spaCy-generated tokens in the corpus and the mean token counts per article–seed-node pair before and after relevant segment selection. Overall, relevant segment selection removed approximately 74% of all tokens. After reconstruction, the mean article lengths were similar across the two languages, suggesting similar levels of coverage within articles in both languages. However, the reduction was greater for English articles, with approximately 82% of tokens removed compared with 68% for Ukrainian articles. This difference can be partly explained by the greater mean length of English articles before selection. Together with the similar average lengths after reconstruction, this suggests that complete English articles tend to contain more information unrelated to their associated seed nodes than complete Ukrainian articles. The average API cost of relevant segment selection was USD 0.0040 per article–seed-node pair.

		English	Ukrainian	Combined
Total tokens	Before selection	1,767,137	2,307,966	4,075,103
	After selection	312,727	743,686	1,056,413
	Reduction	82.30%	67.78%	74.08%
Mean article length	Before selection	1,146.00	684.65	829.45
	After selection	202.81	220.61	215.02

Table 6.6: Total token counts and mean article lengths before and after relevant segment selection.

6.4 Named Entity Recognition

As in the preceding sections, this section first reports the IAA and the composition of the evaluation dataset before presenting the NER evaluation results for two input variants, namely the human-selected input, which consists of text based on the reference annotations from segment selection, and the pipeline-generated input. It then compares performance across source languages and entity types and investigates the observed

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

differences. Finally, it presents the composition of the NER pipeline output by entity type and source language.

6.4.1 Evaluation

Section 5.4.1 describes the construction of the NER evaluation dataset. Table 6.7 reports the IAA for each entity type using micro-F_1. The results indicate high agreement across all entity types, suggesting that the entity definitions and annotation rules were clear to the annotators.

Entity type	Micro-F_1
DATE	0.96
PRODUCT_TYPE	0.90
NAMED_PRODUCT	0.89
GPE	0.92
ORG	0.92

Table 6.7: Inter-annotator agreement for the NER evaluation dataset.

Table 6.8 reports the number of entity mentions in the final reference annotations. Because DATE is comparatively rare, its class-specific evaluation results should be interpreted with caution.

Entity type	Entity mentions
DATE	34
PRODUCT_TYPE	701
NAMED_PRODUCT	301
GPE	195
ORG	507

Table 6.8: Number of entity mentions by type in the final reference annotations.

Following the evaluation procedure described in Section 5.4.2, Table 6.9 summarises the resulting micro-F_1 scores for the human-selected and pipeline-generated inputs. The results indicate consistently high extraction performance across all entity types. For the human-selected input, the F_1 score for every entity type exceeds 0.8 across all runs. PRODUCT_TYPE achieves the lowest performance, which may reflect ambiguity in the span boundaries of product-type mentions. These mentions consist of common nouns that may include modifiers, rather than proper nouns, which are generally easier to distinguish. For the pipeline-generated input, the mean F_1 score for every entity type on the combined dataset remains at least 0.776. This indicates that extraction performance remains high when NER receives the output of the preceding segment-selection stage.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Source language	Entity type	Human-selected input	Pipeline-generated input	ΔF_1
English	DATE	0.929 ± 0.000	0.773 ± 0.037	-0.156
	PRODUCT_TYPE	0.837 ± 0.014	0.778 ± 0.018	-0.059
	NAMED_PRODUCT	0.945 ± 0.016	0.924 ± 0.023	-0.022
	GPE	0.918 ± 0.016	0.851 ± 0.022	-0.067
	ORG	0.946 ± 0.007	0.933 ± 0.011	-0.013
	Overall	0.905 ± 0.006	0.864 ± 0.010	-0.041
Ukrainian	DATE	0.878 ± 0.040	0.781 ± 0.036	-0.098
	PRODUCT_TYPE	0.864 ± 0.008	0.785 ± 0.019	-0.080
	NAMED_PRODUCT	0.909 ± 0.015	0.831 ± 0.014	-0.079
	GPE	0.954 ± 0.008	0.764 ± 0.027	-0.191
	ORG	0.885 ± 0.011	0.837 ± 0.022	-0.048
	Overall	0.886 ± 0.005	0.803 ± 0.017	-0.083
Combined	DATE	0.901 ± 0.022	0.776 ± 0.019	-0.126
	PRODUCT_TYPE	0.854 ± 0.005	0.782 ± 0.012	-0.072
	NAMED_PRODUCT	0.923 ± 0.008	0.866 ± 0.012	-0.057
	GPE	0.935 ± 0.008	0.810 ± 0.020	-0.126
	ORG	0.920 ± 0.006	0.891 ± 0.013	-0.029
	Overall	0.895 ± 0.003	0.831 ± 0.010	-0.064

Table 6.9: NER F_1 scores by article source language and entity type for the human-selected and pipeline-generated inputs. The F_1 scores are reported as mean ± sample standard deviation across ten runs. ΔF_1 is calculated as the pipeline-generated mean minus the human-selected mean.

Another observation beyond the overall scores concerns the difference between Ukrainian-source texts, which had been translated into English by this stage of the pipeline, and English-source texts. With the human-selected input, NER performance is only marginally lower for Ukrainian-source texts. With the pipeline-generated input, however, the performance decline for Ukrainian-source texts is approximately twice as large as that for English-source texts.

A breakdown of the results by metric and source language for the human-selected and pipeline-generated inputs is provided in both Figure 6.4, which shows all evaluation runs, and Table G.1 in Appendix G. Both show that the performance decline is primarily driven by precision. It decreases by 0.13 for Ukrainian-source texts, 0.06 for English-source texts, and 0.10 for the entire dataset. Recall decreases by only 0.02 for English-source texts and 0.03 for both Ukrainian-source texts and the combined dataset. The larger decline in precision is consistent with the preceding segment-selection stage's prioritisation of recall, since entity mentions extracted from false-positive segments are, by construction, counted as false positives in the NER evaluation. However, the small difference in segment-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

selection performance between the source languages cannot explain the substantially larger precision decline for Ukrainian-source texts (see Table 6.5).

Table 6.10 shows the average number of entity mentions extracted per false-positive segment. On average, false-positive Ukrainian-source segments contain more extracted entity mentions than their English-source counterparts. Given the strong NER performance obtained with the human-selected input, this difference is unlikely to result from model errors when processing texts translated from Ukrainian. Instead, it suggests a higher underlying prevalence of entity mentions. Together with the slightly lower segment-selection precision for Ukrainian-source texts, this higher prevalence causes more entity mentions to be counted as false positives, providing a plausible explanation for the larger declines in precision and, consequently, in F_1.

Source language	Predicted mentions per segment
English	0.788 ± 0.138
Ukrainian	1.273 ± 0.141

Table 6.10: Predicted entity mentions in false-positive selected segments by source language, aggregated across all entity types and excluding mentions also found in true-positive segments. Values are reported as mean ± sample standard deviation across ten runs.

6.4.2 Pipeline Named Entity Recognition Results

Table 6.11 reports the number of entity mentions extracted from the pipeline-generated input by entity type and source language. More entity mentions were extracted from Ukrainian-source texts than from English-source texts for every entity type. The average API cost was USD 0.0130 per analysed article–seed-node pair, making NER the most expensive pipeline step.[1]

Entity type	English	Ukrainian	Total
`DATE`	800	1,839	2,639
`PRODUCT_TYPE`	6,581	19,542	26,123
`NAMED_PRODUCT`	2,060	7,177	9,237
`GPE`	2,124	4,143	6,267
`ORG`	6,632	13,217	19,849

Table 6.11: Number of entity mentions extracted by the NER pipeline by entity type and source language, as well as total counts per entity type.

[1] Due to an API configuration error, this average was based on only 2,487 of the full 4,913 article–seed-node pairs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

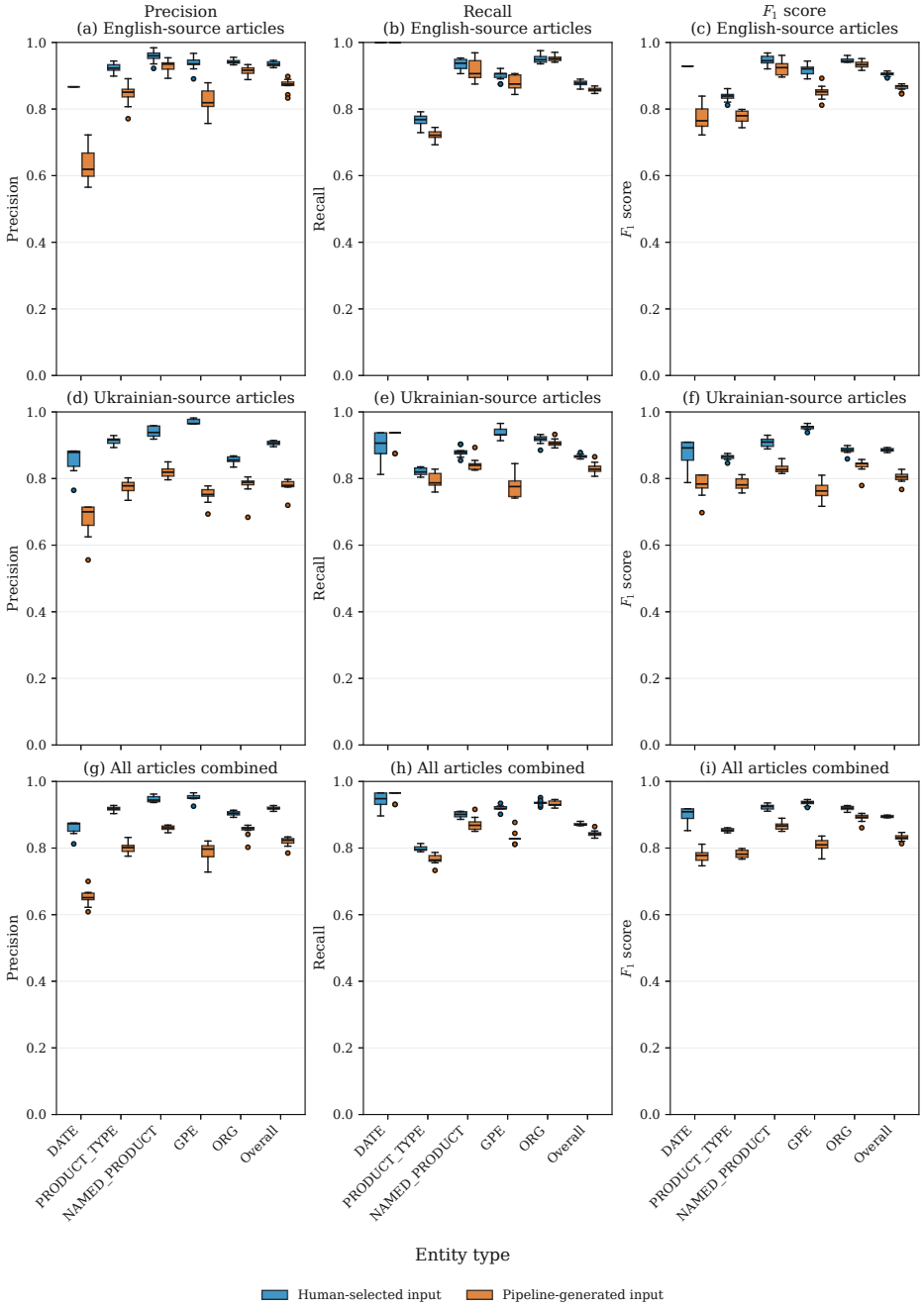

Figure 6.4: NER precision, recall, and F_1 scores across ten evaluation runs for the human-selected and pipeline-generated inputs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6.5 Relationship Extraction

This section presents the results for RE. It first reports the IAA for the alias and relationship annotations and discusses cases of disagreement. As in the NER evaluation, it then presents the evaluation results for the human-selected input, which uses the reference annotations from the preceding two steps, and the pipeline-generated input. Performance is reported separately by relationship type and source language, and the observed differences are interpreted. Finally, it reports the pipeline output by relationship type and source language.

6.5.1 Evaluation

Aliases were annotated alongside the relationships, and the IAA for the alias annotations was 0.58. This score indicates that the alias definitions were difficult for the annotators to apply. After the annotators resolved their disagreements through discussion, the aligned alias reference dataset contained 42 alias groups.

Using these aligned alias groups, the IAA for the relationship annotations created by the two annotators is reported in Table 6.12. Agreement was high for `is_type`, `located_in`, and `makes`, with only small decreases for full relationships, which include annotated attributes. For `supplies_to`, however, the two annotators achieved low agreement when the product attributes essential to this relationship were included. During the discussion, it became clear that the low agreement arose from the difficulty of determining whether a product was supplied directly from a supplying organisation to a receiving organisation or whether an intermediary was involved, as well as which products were involved.

Relationship type	Relationship variant	Micro-F_1
`is_type`	Core	0.827
`located_in`	Core	0.806
	Full	0.774
`makes`	Core	0.853
	Full	0.822
`supplies_to`	Core	0.588
	Full	0.235

Table 6.12: Relationship-annotation IAA by type for the core and full relationship variants, measured using micro-F_1. The core variant excludes attributes, whereas the full variant includes them. Only the core variant applies to `is_type`, which has no attributes.

Table 6.13 shows the number of relationships in the final reference annotations after relationship alignment. The small number of `supplies_to` relationships limits the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

conclusions that can be drawn about the model's performance for this relationship type. Following relationship alignment, four alias groups were removed because they were no longer involved in any annotated relationship.

Relationship type	Relationships	With attributes
`is_type`	100	N/A
`located_in`	32	1
`makes`	203	33
`supplies_to`	10	10

Table 6.13: Number of relationships by type in the final relationship reference annotations, together with the number with attributes. Attributes are not applicable to `is_type`.

The RE evaluation results for full relationships are summarised in Table 6.14, which reports the F_1 scores, and Figure 6.5, which additionally compares precision and recall. Table H.1 in Appendix H presents the complete precision, recall, and F_1 results, broken down by source language and relationship type for core- and full-relationship evaluation.

Source language	Relationship type	Human-selected input	Pipeline-generated input	ΔF_1
English	`is_type`	0.795 ± 0.035	0.686 ± 0.032	-0.108
	`located_in`	0.761 ± 0.044	0.701 ± 0.047	-0.060
	`makes`	0.753 ± 0.033	0.689 ± 0.033	-0.064
	`supplies_to`	0.290 ± 0.226	0.219 ± 0.336	-0.071
	Overall	0.757 ± 0.020	0.681 ± 0.032	-0.076
Ukrainian	`is_type`	0.765 ± 0.021	0.707 ± 0.022	-0.058
	`located_in`	0.827 ± 0.036	0.667 ± 0.044	-0.160
	`makes`	0.750 ± 0.022	0.667 ± 0.029	-0.083
	`supplies_to`	0.142 ± 0.050	0.131 ± 0.043	-0.012
	Overall	0.735 ± 0.009	0.659 ± 0.022	-0.077
Combined	`is_type`	0.776 ± 0.021	0.699 ± 0.021	-0.077
	`located_in`	0.799 ± 0.026	0.684 ± 0.025	-0.115
	`makes`	0.751 ± 0.009	0.676 ± 0.021	-0.075
	`supplies_to`	0.173 ± 0.066	0.150 ± 0.087	-0.023
	Overall	0.744 ± 0.006	0.668 ± 0.019	-0.077

Table 6.14: Full-relationship F_1 scores by article source language and relationship type for the human-selected and pipeline-generated inputs. The F_1 scores are reported as mean $\pm$ sample standard deviation across ten runs. ΔF_1 is calculated as the pipeline-generated mean minus the human-selected mean.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

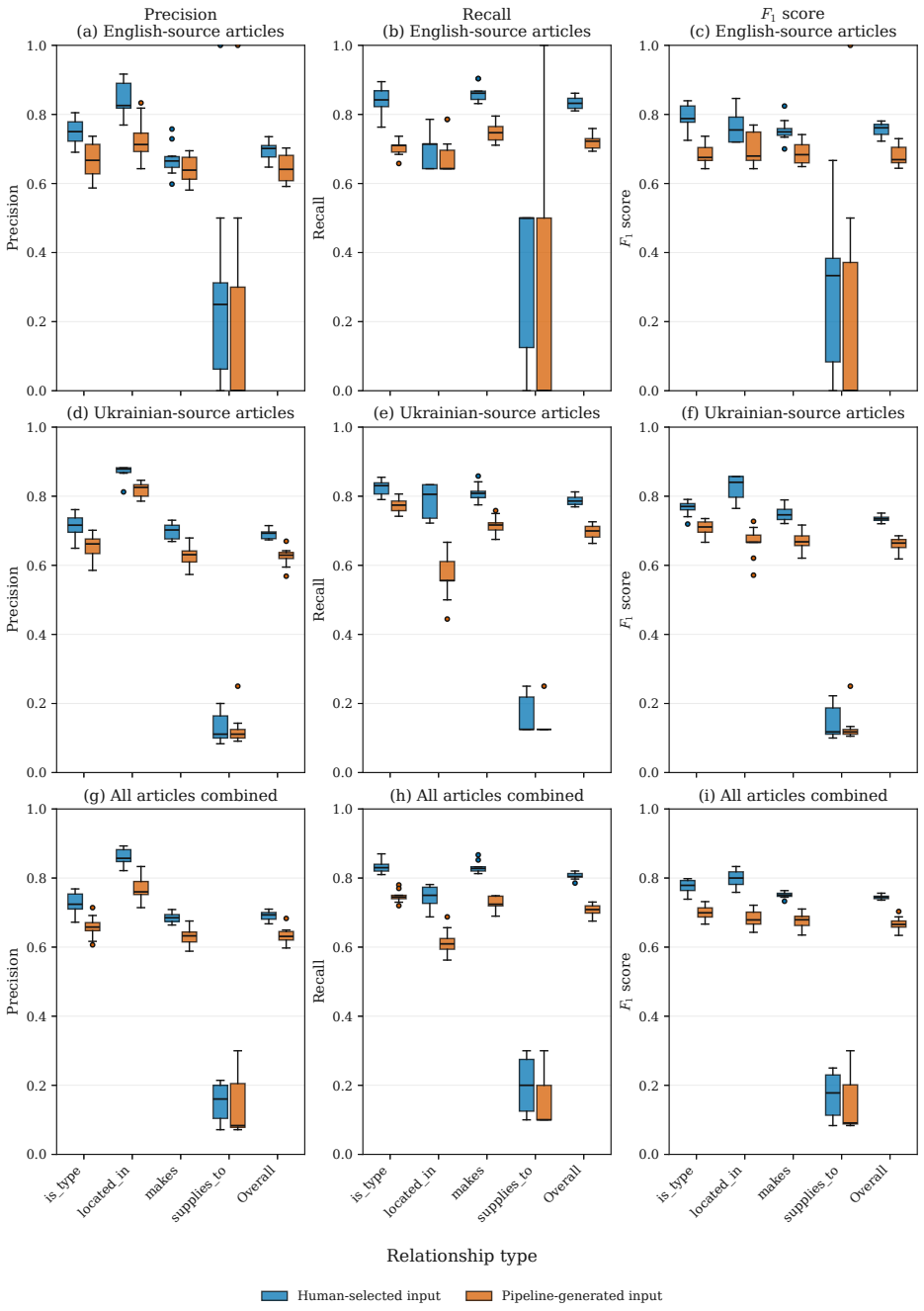

Figure 6.5: RE full-relationship micro-precision, micro-recall, and micro-F_1 scores across ten evaluation runs for the human-selected and pipeline-generated inputs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The comparison of core and full relationships in Table H.1 shows only a slight decrease in overall performance when attributes are included for both the human-selected and pipeline-generated inputs, suggesting that attributes are generally extracted correctly.

At the level of individual relationship types, the figure and tables show that `supplies_to` performs worst and has the highest standard deviation, consistent with the IAA results. Despite this, the pipeline-generated input achieves an overall full-relationship F_1 score of 0.668 ± 0.019 across both languages.

When compared by source language, the F_1 score with the human-selected input was slightly higher for English-source texts than for Ukrainian-source texts, with a difference of approximately 0.02. This mirrors the NER results (see Table 6.9). The performance gap persisted with the pipeline-generated input, for which the F_1 score declined equally for both source languages. Thus, the approximately twice as large decline in NER performance for Ukrainian-source texts, driven primarily by lower precision, did not translate into a larger decline in RE performance. This suggests that the focus of RE on the seed node limited the negative effect of the additional entity mentions supplied by NER, supporting the prioritisation of recall over precision in the earlier pipeline stages.

The evaluation results for alias extraction using the pipeline-generated input are presented in Figure 6.6. The mean F_1 score of 0.441 ± 0.021 was lower than the IAA score and was driven primarily by low precision, indicating that the model extracted many additional incorrect alias groups. This is critical because using these groups to resolve entity mentions during knowledge-graph construction could create many factually incorrect links. We therefore decided not to use the predicted alias groups when constructing the knowledge graph.

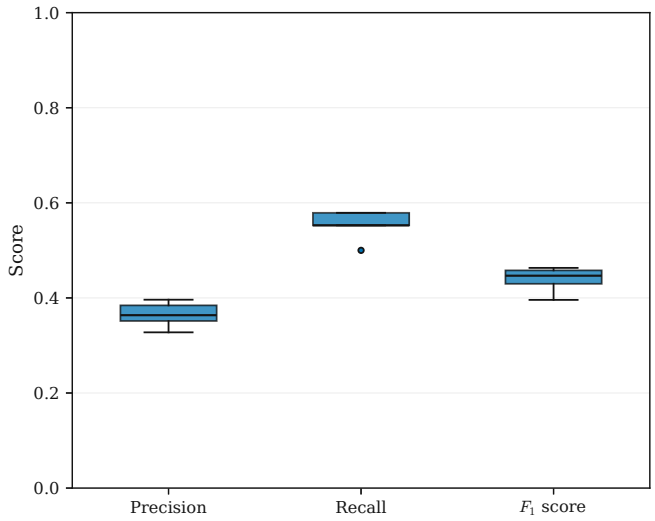

Figure 6.6: Alias-extraction precision, recall, and F_1 scores across ten evaluation runs for the pipeline-generated input.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6.5.2 Pipeline Relationship Extraction Results

Despite its comparatively weak evaluation performance, `supplies_to` was retained alongside the other relationship types during pipeline extraction. Using the entity mentions and text produced by the preceding pipeline stages, the pipeline extracted 18,866 relationship instances in total. Table 6.15 reports these instances by relationship type and source language, together with the overall counts. These counts represent raw relationship extractions before entity resolution and relationship deduplication. The average API cost was USD 0.0074 per article–seed-node pair.

Relationship type	English	Ukrainian	Total
`is_type`	1,099	4,181	5,280
`located_in`	508	907	1,415
`makes`	2,727	8,483	11,210
`supplies_to`	188	773	961

Table 6.15: Number of relationship instances extracted by relationship type and source language, as well as total counts per relationship type.

6.6 Knowledge Graph

In the final section of the Results chapter, the structure and composition of the constructed supply-chain knowledge graph are first described. The temporal dimension of the knowledge graph is then explored, followed by an analysis of relationship coverage by source language.

6.6.1 Knowledge Graph Structure

Following entity resolution, 11,213 relationship instances were identified as duplicates and removed. The remaining relationships were then used to construct the knowledge graph. Figure 6.7 illustrates the types of information represented in the graph through an example subgraph and indicates the languages in which the relationships were identified. It shows a supply relationship between The Fourth Law and Vyriy, both of which were seed nodes in this use case. The Fourth Law makes TFL-1, an autonomous guidance system for drones, and supplies it to Vyriy. The inclusion of TFL-1 in the name Vyriy-10-TFL-1 suggests that Vyriy integrates the system into this FPV drone. Both companies are located in Kyiv, and The Fourth Law also appears to have a presence in the United States.

Tables 6.16 and 6.17 summarise the composition of the final knowledge graph at different minimum evidence counts. The evidence count is the number of articles from which a relationship or entity was extracted, with higher counts reflecting greater confidence. The degree was calculated using an undirected version of the knowledge graph, in which each

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

SUPPLY node was replaced by an edge between the supplier and recipient. Increasing the minimum evidence count from one to three reduced the number of nodes by approximately 80% and the number of relationships by approximately 84%. Consequently, more than 80% of the relationships were identified in only one or two articles.

At a minimum evidence count of three, the mean degrees of ORG and NAMED_PRODUCT entities were lower than at a minimum evidence count of one, whereas those of PRODUCT_TYPE and GPE entities were higher. At both evidence counts, however, ORG nodes had the highest mean degree. This is expected because organisations can participate in three of the four defined relationship types. Overall, the observed mean degrees appear plausible given the knowledge-graph schema. For example, a NAMED_PRODUCT is generally linked to one manufacturer but may be assigned to multiple product types, such as both a reconnaissance drone and a strike drone.

Figure 6.7: Example subgraph at a minimum evidence count of one. Node colours indicate entity type. Pink nodes represent PRODUCT_TYPE entities, red nodes NAMED_PRODUCT entities, green nodes ORG entities, and blue nodes GPE entities. Yellow nodes are intermediate SUPPLY nodes representing supplies_to relationships. The colour of each relationship indicates the type of its source node. Ukrainian and British flags indicate relationships identified in Ukrainian- and English-language news articles, respectively.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Evidence count	is_type	located_in	makes	supplies_to	Total
1	2,238	234	3,555	1,626	7,653
3	390	69	670	101	1,230

Table 6.16: Number of distinct knowledge-graph relationships by type and in total at minimum evidence counts of one and three. Each SUPPLY node is counted as one supplies_to relationship.

Evidence count	Metric	ORG	NAMED_PRODUCT	PRODUCT_TYPE	GPE	Overall
1	Node count	536	627	1,419	55	2,637
	Mean degree	13.14	4.88	3.50	4.25	5.80
3	Node count	117	203	194	15	529
	Mean degree	8.04	2.99	4.35	4.60	4.65

Table 6.17: Entity-node counts and mean degree by entity type and minimum evidence count. Degree is calculated after replacing each SUPPLY node with a supplier–recipient edge.

6.6.2 Temporal Knowledge Graph Composition

Figure 6.8 shows the numbers of new and recurring distinct entities and relationships for four two-year intervals from 2018 to 2025. Interval assignments were based on extracted year mentions, with article capture dates used when no year mention was available. An element was considered recurring if it had appeared in an earlier interval. Only relationships supported by at least three articles were included to increase confidence in the observed temporal patterns.

Counts generally increased between consecutive intervals, except for supplies_to in 2020–2021. The increase was especially large in 2022–2023, the first interval following the start of the full-scale invasion. For every type, the number of new elements in this interval exceeded the corresponding total in both earlier intervals. In the final interval, the number of new elements decreased for every entity and relationship type except NAMED_PRODUCT, is_type, and makes, for which it continued to increase. The decline in newly observed ORG and PRODUCT_TYPE entities suggests that the establishment of organisations and the introduction of broader product types were more concentrated in the first two years of the war. In contrast, the continued growth in NAMED_PRODUCT entities, as well as in is_type relationships, indicates that organisations continued to release new products during the final interval examined.

Using the same two-year intervals and distinction between new and recurring relationships, Figure 6.9 provides a more detailed breakdown of makes relationships involving selected drone product types. No makes relationships involving FPV drones were identified before the full-scale invasion, but after its start, their number increased from 14 in 2022–2023 to 43 in 2024–2025. Relationships involving interceptor drones appear only in the final

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

interval. In contrast, the numbers of relationships involving strike drones, reconnaissance drones, and loitering munitions increased across all four intervals. Relationships involving cargo and agricultural drones, which are associated with more civilian applications, appear only in the two intervals following the start of the full-scale invasion.

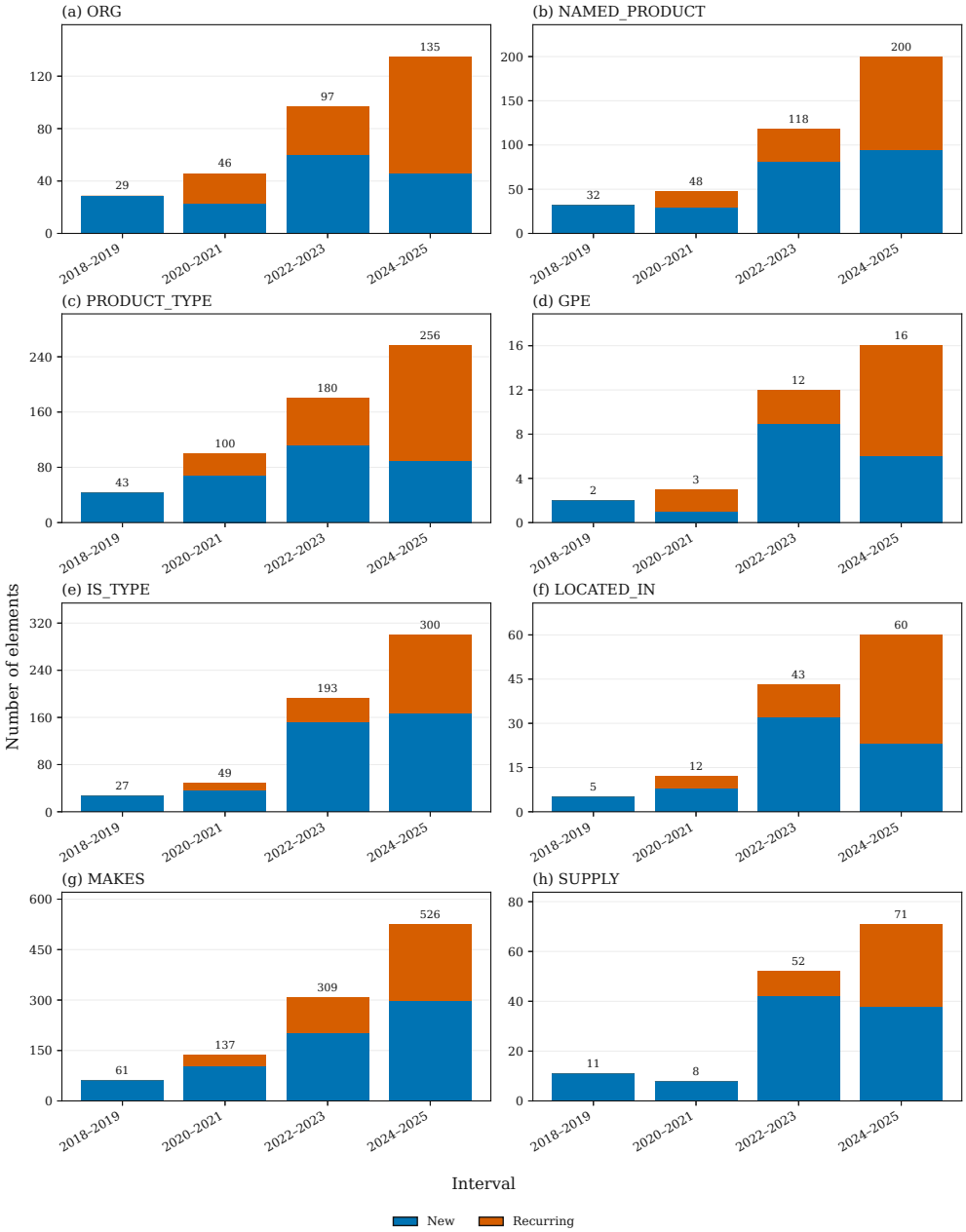

Figure 6.8: New and recurring distinct entities and relationships by type and two-year interval at a minimum evidence count of three.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

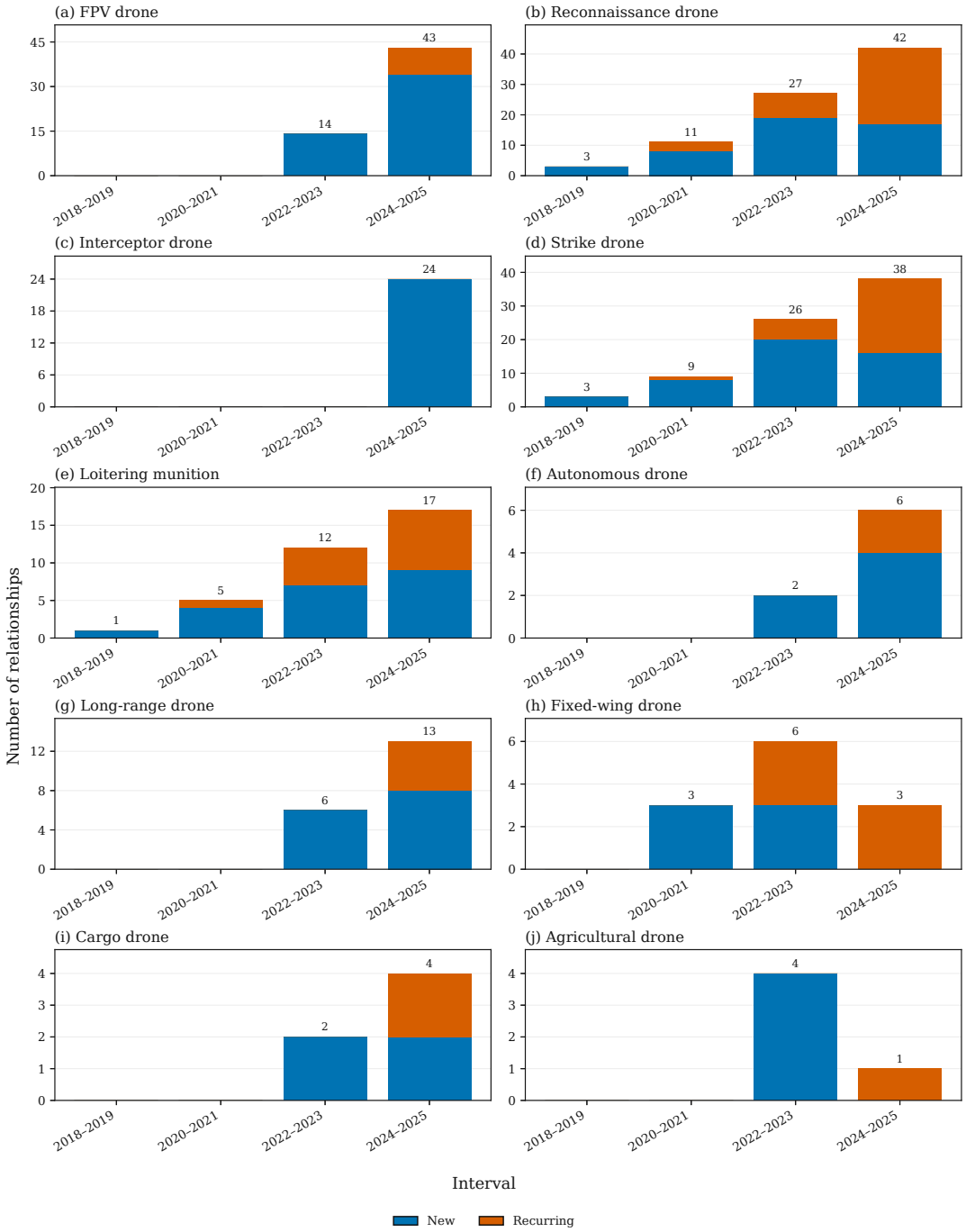

Figure 6.9: New and recurring distinct `makes` relationships by drone type and two-year interval at a minimum evidence count of three.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6.6.3 Relationship Coverage by Source Language

Figure 6.10 shows the source-language coverage of relationships in the knowledge graph. In contrast to the temporal analysis, all extracted relationships were included, regardless of the number of supporting articles, to show the source-language distribution of all relationships in the knowledge graph. It reports the share of relationships identified only in Ukrainian-language articles, only in English-language articles, or in both. More than 60% of the relationships would be absent from the knowledge graph if the source corpus were restricted to English-language articles. The share identified only in English-language articles is relatively consistent across relationship types and accounts for approximately 20% of all relationships. Total English-language coverage, comprising relationships identified either only in English-language articles or in both languages, accounts for approximately 30%.

Figure 6.7 also reveals that if the knowledge graph were constructed using only English-language articles, neither the classification of TFL-1 as an autonomous guidance system nor The Fourth Law's location in the United States would be represented.

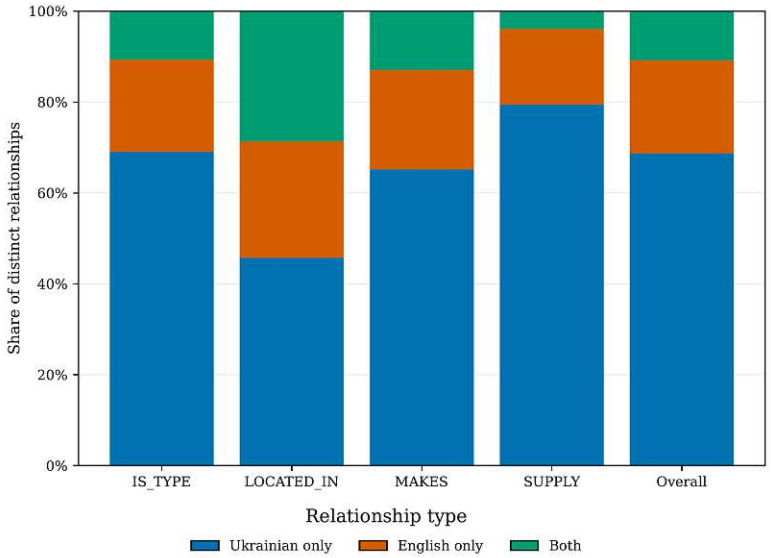

Figure 6.10: Language composition of distinct relationships by relationship type at a minimum evidence count of one.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 7

Discussion and Conclusion

In this final chapter, we first assess the suitability of our extraction pipeline for producing a supply-chain knowledge graph that can be used to answer our research questions. We do so by discussing the evaluation results for its individual steps and identifying the pipeline's limitations. We then provide answers to the research questions before discussing in more detail the limitations of our approach beyond the pipeline and potential directions for future research. Finally, we conclude the thesis.

7.1 Knowledge-Graph Extraction Pipeline

Answering the research questions requires a suitable information retrieval and extraction pipeline for processing the keyword-filtered, preprocessed CC-News corpus in English and Ukrainian. To assess its suitability for multilingual analysis, the performance and limitations of the individual retrieval and information extraction stages are discussed both overall and separately by the source language of the articles. This assessment is based on manually annotated evaluation datasets constructed specifically for this use case involving the drone supply chain.

The first pipeline stage considered is document retrieval, which uses separate retrievers for English and Ukrainian articles. The evaluation results for both retrievers (see Section 6.1) do not suggest that they miss a substantial number of relevant articles containing supply-chain information and instead indicate good retrieval performance. However, the evaluation set, which was constructed to contain many relevant article–seed-node pairs, covers only a subset of the corpus. Therefore, the results should be interpreted as an indication of retrieval performance rather than as an absolute statement about performance on the full corpus.

The evaluation results for the subsequent relevance classifier show strong performance, with almost no difference between the two source languages, and therefore raise no

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

concerns regarding pipeline quality (see Section 6.2).

Segment extraction constitutes the next stage of the pipeline and was designed to achieve high recall while maintaining acceptable precision, thereby minimising the loss of relevant passages. Its evaluation results met this objective for both source languages, although performance was slightly lower for Ukrainian-source articles (see Section 6.3). Nevertheless, further improvements in precision would likely benefit downstream performance. Moreover, the potential negative effects of segment extraction on downstream performance were not evaluated, as removed segments may contain context needed to extract relationships.

After relevant segment selection, the NER and RE stages processed only English text because Ukrainian-source articles had been translated into English. Since we did not evaluate translation quality, we cannot determine whether relationship information was lost during translation. Our analysis therefore relies on the assumption that the translations were accurate and that no relationship information was lost.

Under this assumption, we can place our NER evaluation results presented in Section 6.4 in context by considering the results reported by AlMahri et al. [12], whose closely related approach uses LLMs to extract supply-chain relationships from news articles. Although the results are not directly comparable because the evaluation datasets, models, and definitions of relationships and entities differ, they provide a reference point for performance in the field. They reported an overall F_1 score of 0.520, whereas our pipeline achieved 0.831 ± 0.010. This comparison indicates that our NER approach performs sufficiently well for our use case. Regarding language-specific performance, we found that NER performed considerably better on pipeline-generated input from English-source articles than on input from Ukrainian-source articles. This difference could be explained by the greater number of entity mentions in false-positive segments extracted from Ukrainian-source text.

AlMahri et al.'s results also provide a point of comparison for the final information-extraction stage, RE [12]. They reported an F_1 score of 0.328, whereas our pipeline achieved 0.668 ± 0.019 (see Section 6.5). This comparison indicates that our approach achieves reasonable performance within the supply-chain domain. Nevertheless, our evaluation results suggest that many relationships were incorrectly extracted or not detected. Regarding source-language performance, the difference observed for NER was reduced at the RE stage. Performance remained only slightly lower for Ukrainian-source articles.

The conservative entity-resolution approach described in Section 4.6.1 was designed to reduce false positives by avoiding incorrect merges between mentions of different entities. However, its emphasis on precision can produce false negatives by leaving mentions of the same entity unresolved, preventing equivalent relationships from being deduplicated and potentially inflating their counts.

Overall, the evaluation results suggest that the pipeline can support analyses of temporal patterns and source-language differences in relationship coverage within the extracted

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

knowledge graph. Nevertheless, errors in information extraction and entity resolution limit these analyses to identifying general trends and directions rather than supporting precise quantitative conclusions about the underlying supply chain.

7.2 Answers to the Research Questions

Which temporal trends in the growth of Ukraine's UAV supply chain since the start of the full-scale invasion are reflected in the knowledge graph constructed from multilingual news articles?

To answer this research question, the results presented in Section 6.6.2 are compared with documented real-world developments. First, we observed a significant increase in the number of entities and relationships across both two-year intervals following the start of the full-scale invasion in 2022 compared with the preceding four years, with growth continuing from the first post-invasion interval to the second. This pattern aligns with the substantial expansion of Ukraine's pre-existing but small drone industry following the full-scale invasion and its continued growth throughout the war [49], [50], [51], [52].

The drone-type analysis suggests that the data capture both the presence and emergence of producers of specific drone types. For example, `makes` relationships identifying producers of interceptor drones appear in our data only from 2024 onward, consistent with the real-world emergence of interceptor drones [53], [54]. The knowledge graph also contains `makes` relationships identifying producers of reconnaissance drones before the start of the full-scale invasion. This is consistent with the real-world example of UkrSpecSystems, which developed the PD-1 and PD-2 before the war [55], [56].

Limitations are also evident. For example, producers of civilian drones such as agricultural drones were already operating in Ukraine before the start of the full-scale invasion [57], [58]. However, we did not observe any `makes` relationships identifying producers of these drone types during that period. Their absence may be a result of incomplete seed-node coverage, relevant articles not being available in the CC-News corpus, available articles not being identified by the retrieval stage, relevant relationships being missed during relationship extraction, temporal information about relationships being stated in relative terms (e.g., "before the war"), or entity resolution failing to identify synonyms for agricultural drones. This indicates that, although the knowledge graph represents many temporal trends and phenomena well (e.g., the emergence of interceptor drones and their producers), it cannot yet provide a complete and accurate reconstruction of temporal developments in Ukrainian drone supply chains.

Overall, these findings show that the knowledge graph constructed by our method captures the general growth trends of Ukraine's drone industry. The product-type analysis further shows that it reflects specific developments both before and during the full-scale invasion. Nevertheless, the absence of `makes` relationships involving agricultural drones highlights that the reconstruction remains incomplete.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

What difference in relationship coverage arises when the knowledge graph is constructed from only English-language articles rather than from both English- and Ukrainian-language articles?

The results presented in Section 6.6.3 answer this research question. Within the limitations outlined in the preceding section, they suggest that restricting the corpus to English-language articles would omit more than half of the relationships represented in the multilingual knowledge graph. Consequently, an English-only knowledge graph would provide incomplete coverage of the relationships defined in this thesis for Ukraine's UAV supply chain. Beyond our use case, this finding indicates that restricting news-based supply-chain extraction to English can leave relevant supply-chain information undiscovered.

7.3 Limitations

In addition to the limitations already discussed in the Knowledge-Graph Extraction Pipeline section, the following limitations apply.

The seed nodes do not represent the entire drone industry in 2026, and the aliases collected for the included seed nodes are unlikely to be complete.

Another limitation arises from the limited development of the NER and RE pipeline stages, which resulted in only a small set of entity and relationship types being defined. These types do not fully capture the entities and relationships involved in complex supply chains and exclude, for example, ownership relationships.

The pipeline evaluation presents additional limitations. Only one method and configuration were evaluated for each information-extraction stage, with no evaluation of alternative methods or pipeline designs. Therefore, it remains unclear whether the selected configurations were optimal, whether selecting segments negatively affected subsequent performance, and whether performing NER and RE in two separate steps rather than one improved or harmed performance. Additionally, pipeline performance was not evaluated separately across time periods, so differences between earlier and later articles cannot be quantified. Entity resolution was also not evaluated, which not only prevents an assessment of its performance but also limits the basis for future improvements.

Beyond these pipeline-specific limitations, we examined only one use case and therefore cannot determine whether our method can extract temporal patterns in other use cases. The analysis was also limited to the articles available through CC-News, which did not include all relevant sources, such as `https://militarnyi.com/en/`. Furthermore, no credibility ranking was applied to the news sources, and all sources were considered equally. Consequently, relationships may have been extracted from articles even though the underlying claims were factually incorrect.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

7.4 Future Work

Based on these limitations, several directions for future work are proposed.

Alternative methods and configurations should be evaluated for each pipeline stage. For example, GLiNER2 [59] could be assessed for NER. Future work should also determine whether segment selection improves performance and whether NER and RE can be performed jointly. To improve entity resolution, more sophisticated approaches, such as methods based on semantic similarity or LLMs, should be evaluated.

The knowledge graph could also be extended with additional entity and relationship types to provide a more comprehensive representation of the supply chain. For example, a `part_of` relationship could link products, while a `used_by` relationship could indicate that an organisation uses a product without implying a direct supply relationship.

Another direction would be to run the pipeline iteratively rather than only once, as in this thesis. This could reveal supply relationships beyond the first tier and provide insights into deeper levels of the supply chain.

Future research should also assess the method beyond the present use case by examining whether it can extract temporal supply-chain patterns in other domains that have experienced growth, such as semiconductors or green-energy technologies. For such applications, additional news sources should be integrated to provide a more complete picture of the underlying supply chains.

7.5 Conclusion

In this thesis, we constructed a knowledge graph from Ukrainian- and English-language news articles obtained from the large-scale multilingual CC-News corpus. We used this graph to investigate whether temporal developments in a supply chain can be reconstructed through relationship extraction and to assess the importance of incorporating multilingual text sources into the extraction process.

We found that general temporal trends, such as the growth of the drone industry, can be reconstructed. Moreover, for selected drone types, we found that the graph can capture both producers that emerged after the full-scale invasion and those already present before it. However, for other drone types, such as agricultural drones, our approach failed to detect that the industry was already present before the war, demonstrating its limitations. The language analysis showed that including Ukrainian-language articles is critical because these articles accounted for a large share of the relationship coverage.

To conclude, within the scope of our use case, these findings show the importance of multilingual data sources and indicate that, while it is possible to extract a supply-chain graph representing general temporal trends from CC-News, further work is needed to improve the completeness of the graph.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Overview of Generative AI Tools Used

OpenAI Codex (https://openai.com/codex/) was used with OpenAI's GPT-5.6 Sol (https://openai.com/index/previewing-gpt-5-6-sol/) and GPT-5.5 (https://openai.com/index/introducing-gpt-5-5/) models to revise text already written by the author by correcting grammar and spelling mistakes and improving sentence structure, clarity, transitions, and wording. It also provided feedback on consistency and structure across the thesis. Codex further supported the creation of LaTeX tables from existing data and the formatting of tables, figures, and formulas, as well as general LaTeX formatting. It assisted the author in creating the pseudocode used to present algorithms and in implementing the code for the plots in this thesis.

ChatGPT (https://chatgpt.com/) was used with GPT-5.6, GPT-5.5, and GPT-5.4 (https://openai.com/index/introducing-gpt-5-4/) to assist with literature searches. Each suggested source was independently reviewed by the author before inclusion. ChatGPT and Codex also assisted the author in creating BibTeX entries. Both tools provided assistance with the implementation, debugging, refactoring, and code review of the data-preprocessing pipelines, the knowledge-graph construction pipeline, and the scripts used to evaluate the knowledge-graph construction pipeline, as well as with the modification of the Prodigy interface and the development of the custom tool used to align relationship annotations.

The DeepL Translator (https://www.deepl.com/translator) was used to translate the acknowledgements from German into English. It was also used more generally to translate English text into Ukrainian and Ukrainian text into English during the development of the thesis.

DeepL Write (https://www.deepl.com/de/write) was occasionally used to correct grammar and spelling mistakes and improve sentence structure, clarity, transitions, and wording in text already written by the author.

All outputs resulting from the AI assistance described above were critically evaluated and reviewed by the author and were edited when necessary.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Figures

2.1 Probability of classifying two sets as candidate duplicates as a function of Jaccard similarity for three different configurations. 10

3.1 CC-News main-text extraction pipeline implemented with DataTrove. . . 14
3.2 Overview of the seed-node and quality filtering pipeline. 16
3.3 Composition of the preprocessed corpus after deduplication by (a) seed node, (b) language, and (c) capture month. Articles associated with multiple seed nodes are counted once for each seed node in (a). 19

4.1 Overview of the pipeline used to construct the knowledge graph based on the preprocessed corpus. 22
4.2 Example single-query and multi-query outputs generated for Q2 on 15 May 2026 using GPT-5.5 with reasoning set to medium. The Q2 keyword query was passed as user input to the respective prompts provided in Appendix I.2. 27
4.3 Application of the segment-relevance criteria to an excerpt from a news article published by Deutsche Welle [41] concerning the seed node Skyeton. Light-green segments are considered relevant, while grey segments are considered non-relevant. The criterion labels indicate which criteria apply. 32
4.4 Representation of a supply relationship using an intermediate SUPPLY node. 40

5.1 Modified Prodigy interface displaying linkable entity mentions and relationship-attribute annotation inputs for an excerpt from a Tech.eu news article [48]. 54
5.2 Interface of the custom relationship-disagreement alignment tool displaying excerpts from a Tech.eu news article [48]. 55

6.1 Pairwise confusion matrices for the relevance judgements on the 200 overlapping article–seed-node pairs. Matrix (a) compares the two annotators. Matrices (b) and (c) compare GPT-5 mini with each annotator. 58
6.2 $\overline{\text{Recall@}k}$ across retrieval cut-offs on the development split of SilverFinal for the best-performing configuration of each method and language. . . . 59
6.3 Composition of the 4,340 distinct articles with at least one positive relevance judgement by (a) seed node, (b) language, and (c) capture month. Articles classified as relevant for multiple seed nodes are counted once for each seed node in (a). 62

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

6.4 NER precision, recall, and F_1 scores across ten evaluation runs for the human-selected and pipeline-generated inputs. 67
6.5 RE full-relationship micro-precision, micro-recall, and micro-F_1 scores across ten evaluation runs for the human-selected and pipeline-generated inputs. 70
6.6 Alias-extraction precision, recall, and F_1 scores across ten evaluation runs for the pipeline-generated input. 71
6.7 Example subgraph at a minimum evidence count of one. Node colours indicate entity type. Pink nodes represent `PRODUCT_TYPE` entities, red nodes `NAMED_PRODUCT` entities, green nodes `ORG` entities, and blue nodes `GPE` entities. Yellow nodes are intermediate `SUPPLY` nodes representing `supplies_to` relationships. The colour of each relationship indicates the type of its source node. Ukrainian and British flags indicate relationships identified in Ukrainian- and English-language news articles, respectively. 73
6.8 New and recurring distinct entities and relationships by type and two-year interval at a minimum evidence count of three. 75
6.9 New and recurring distinct `makes` relationships by drone type and two-year interval at a minimum evidence count of three. 76
6.10 Language composition of distinct relationships by relationship type at a minimum evidence count of one. 77

F.1 Number of retained articles by capture month for each of the nine seed nodes with the largest total number of associated articles. All remaining seed nodes are aggregated as Other seed nodes. An article associated with several seed nodes contributes to each relevant series. The dashed line marks the start of the full-scale invasion on 24 February 2022. 118
F.2 Number of retained articles associated with each seed node after deduplication. 119

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Tables

2.1 Examples of Damerau–Levenshtein distances and the edit operations used to transform each source string into its target string. 11

3.1 Schematic representation of two article–seed-node pairs associated with the same article. 18

4.1 English and Ukrainian keyword groups used to construct the keyword queries. GPT-5.5 generated the Ukrainian keywords on 14 May 2026 with reasoning set to medium, using the system prompt in Appendix I.1 and the respective keyword-group descriptions in Table I.1. 25
4.2 Keyword-group combinations defining queries Q1–Q8. The complete keyword groups are provided in Table 4.1. 26
4.3 Entity definitions and examples used in the NER pipeline step. 35
4.4 Relationship definitions and their attributes, source entity types, and target entity types used for RE. 36
4.5 Examples of representative–candidate pairs, whether they are matched by the entity-resolution algorithm, and explanations for non-matches. The italicised text shows the normalised form. Tokens are formed by splitting the normalised form at whitespace. 38
4.6 Selected properties relevant to knowledge-graph analysis. 39

5.1 Overview of sparse retrieval configurations. 42
5.2 Overview of dense retrieval configurations. 42
5.3 Overview of hybrid retrieval configurations. Sparse and dense options shown on the same row are paired. 43
5.4 Sampling weights by language and SilverFinal label. 48

6.1 IAA metrics for the annotations used to validate the SilverFinal dataset. 57
6.2 Weighted evaluation metrics for the relevance labels in SilverFinal by language. 58
6.3 Selected retriever configurations for English and Ukrainian. 59
6.4 Performance of the relevance classifier over ten runs by language and across both languages. Values are reported as mean ± sample standard deviation. 60
6.5 Performance of relevant segment selection over ten runs by language and across both languages. Values are reported as mean ± sample standard deviation. 63

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

6.6 Total token counts and mean article lengths before and after relevant segment selection. 63
6.7 Inter-annotator agreement for the NER evaluation dataset. 64
6.8 Number of entity mentions by type in the final reference annotations. . . 64
6.9 NER F_1 scores by article source language and entity type for the human-selected and pipeline-generated inputs. The F_1 scores are reported as mean ± sample standard deviation across ten runs. ΔF_1 is calculated as the pipeline-generated mean minus the human-selected mean. 65
6.10 Predicted entity mentions in false-positive selected segments by source language, aggregated across all entity types and excluding mentions also found in true-positive segments. Values are reported as mean ± sample standard deviation across ten runs. 66
6.11 Number of entity mentions extracted by the NER pipeline by entity type and source language, as well as total counts per entity type. 66
6.12 Relationship-annotation IAA by type for the core and full relationship variants, measured using micro-F_1. The core variant excludes attributes, whereas the full variant includes them. Only the core variant applies to `is_type`, which has no attributes. 68
6.13 Number of relationships by type in the final relationship reference annotations, together with the number with attributes. Attributes are not applicable to `is_type`. 69
6.14 Full-relationship F_1 scores by article source language and relationship type for the human-selected and pipeline-generated inputs. The F_1 scores are reported as mean ± sample standard deviation across ten runs. ΔF_1 is calculated as the pipeline-generated mean minus the human-selected mean. 69
6.15 Number of relationship instances extracted by relationship type and source language, as well as total counts per relationship type. 72
6.16 Number of distinct knowledge-graph relationships by type and in total at minimum evidence counts of one and three. Each `SUPPLY` node is counted as one `supplies_to` relationship. 74
6.17 Entity-node counts and mean degree by entity type and minimum evidence count. Degree is calculated after replacing each `SUPPLY` node with a supplier–recipient edge. 74

A.1 Ukrainian drone manufacturers and their aliases (English and Ukrainian) used as seed nodes. 101

B.1 Parameter space used to tune the quality filter. 106
B.2 Selected quality-filter parameters and reasons for their selection. 106

G.1 Named entity recognition evaluation results by article source language and entity type, including results for both languages combined, for the human-selected and pipeline-generated inputs. Overall values are micro-averaged. All values are reported as mean ± sample standard deviation across ten runs. 122

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

H.1 Relationship-extraction evaluation results by article source language, relationship type, and relationship variant for the human-selected and pipeline-generated inputs, including results for both languages combined. All values are reported as mean ± sample standard deviation across ten runs. The full relationship variant is not applicable to `is_type`. 124

I.1 Descriptions of the keyword groups provided as user input when translating English keywords into Ukrainian. 125

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Algorithms

C.1 Token-level seed-node matching 108
C.2 Token-level seed-node matching with candidate-token position tracking 109

D.1 Partitioning preliminary clusters by seed-node IDs 111
D.2 Selecting a representative by article length 112
D.3 Validating candidate similarity to the representative 113
D.4 Reclustering rejected candidates and combining the results 114

E.1 SilverExtension sampling for one language 115

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

[1] C. Diem, A. Borsos, T. Reisch, J. Kertész, and S. Thurner, "Quantifying firm-level economic systemic risk from nation-wide supply networks", *Sci. Rep.*, vol. 12, no. 1, May 2022, Art. no. 7719. DOI: 10.1038/s41598-022-11522-z

[2] P. Wichmann, A. Brintrup, S. Baker, P. Woodall, and D. McFarlane, "Towards automatically generating supply chain maps from natural language text", *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1726–1731, 2018, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018. DOI: 10.1016/j.ifacol.2018.08.207

[3] T. Reisch, A. Borsos, and S. Thurner, "Supply chain network rewiring dynamics at the firm level", *PNAS Nexus*, vol. 5, no. 4, Apr. 2026, Art. no. pgag091. DOI: 10.1093/pnasnexus/pgag091

[4] G. Zelbi, L. N. Ialongo, and S. Thurner, "Systemic risk mitigation in supply chains through network rewiring", *Sci. Rep.*, vol. 16, no. 1, Mar. 2026, Art. no. 12334. DOI: 10.1038/s41598-026-42549-1

[5] E. Dhyne, G. Magerman, and S. Rubínová, "The Belgian production network 2002–2012", National Bank of Belgium, Brussels, NBB Working Paper 288, 2015. [Online]. Available: https://hdl.handle.net/10419/144500

[6] T. Reisch, G. Heiler, C. Diem, P. Klimek, and S. Thurner, "Monitoring supply networks from mobile phone data for estimating the systemic risk of an economy", *Sci. Rep.*, vol. 12, no. 1, Aug. 2022, Art. no. 13347. DOI: 10.1038/s41598-022-13104-5

[7] S. Köse et al., "Reconstructing temporal multi-relational firm networks at scale using large language models: The case of the semiconductor industry", 2026. arXiv: 2605.15842.

[8] P. Wichmann, A. Brintrup, S. Baker, P. Woodall, and D. McFarlane, "Extracting supply chain maps from news articles using deep neural networks", *Int. J. Prod. Res.*, vol. 58, no. 17, pp. 5320–5336, 2020. DOI: 10.1080/00207543.2020.1720925

[9] J. Kim, E. Kim, D. Kim, Y. Kim, and T. Cheong, "Extracting supply chain information from news articles using large language models: A fully automatic approach", *IEEE Access*, vol. 13, pp. 156 203–156 214, 2025. DOI: 10.1109/ACCESS.2025.3605837

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[10] T. Liu and H. Meidani, "Supply chain network extraction and entity classification leveraging large language models", in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 3448–3455. DOI: 10.1109/BigData62323.2024.10826017

[11] B. Jin, Q. Sun, and L. Chen, "Enhancing supply chain transparency in emerging economies using online contents and LLMs", in *2025 International Conference on Information Networking (ICOIN)*, 2025, pp. 487–492. DOI: 10.1109/ICOIN63865.2025.10993099

[12] S. AlMahri, L. Xu, and A. Brintrup, "Enhancing supply chain visibility with knowledge graphs and large language models", *Int. J. Prod. Res.*, vol. 64, no. 6, pp. 2178–2209, 2026. DOI: 10.1080/00207543.2025.2575841

[13] I. Jackson, M. J. Saénz, D. Ivanov, and B. J. Ma, "Supply chain mapping through retrieval-augmented generation: Applications to the electronics industry", *J. Oper. Res. Soc.*, vol. 77, no. 7, pp. 1798–1818, 2026. DOI: 10.1080/01605682.2025.2608868

[14] M. Sauser, "Constrained innovation: Drones and the Russo-Ukrainian war", *Survival*, vol. 68, no. 1, pp. 127–148, 2026. DOI: 10.1080/00396338.2026.2620295

[15] A. Broder, "On the resemblance and containment of documents", in *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, 1997, pp. 21–29. DOI: 10.1109/SEQUEN.1997.666900

[16] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 3rd ed. Cambridge University Press, 2020. DOI: 10.1017/9781108684163 Accessed: Aug. 29, 2026. [Online]. Available: http://infolab.stanford.edu/~ullman/mmds/book0n.pdf

[17] M. Honnibal, I. Montani, et al., *Prodigy: An annotation tool for AI, machine learning & NLP*, version 1.18.2, Explosion, Jun. 4, 2025. Accessed: Aug. 29, 2026. [Online]. Available: https://prodi.gy

[18] Prodigy. "Annotation metrics", Prodigy, Accessed: Aug. 29, 2026. [Online]. Available: https://prodi.gy/docs/metrics

[19] R. Artstein, "Inter-annotator agreement", in *Handbook of Linguistic Annotation*, N. Ide and J. Pustejovsky, Eds. Dordrecht: Springer Netherlands, 2017, pp. 297–313. DOI: 10.1007/978-94-024-0881-2_11

[20] K. L. Gwet, "On Krippendorff's alpha coefficient", Advanced Analytics, LLC, Tech. Rep., 2015. [Online]. Available: https://www.agreestat.com/assets/paper-on-krippendorffs-alpha.pdf

[21] K. Krippendorff. "Computing Krippendorff's alpha-reliability", University of Pennsylvania, 2011, Accessed: Aug. 29, 2026. [Online]. Available: https://repository.upenn.edu/entities/publication/034a6030-c584-4d14-9d3d-7b7e8d16df20

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[22] F. J. Damerau, "A technique for computer detection and correction of spelling errors", *Commun. ACM*, vol. 7, no. 3, pp. 171–176, Mar. 1964. DOI: 10.1145/363958.363994

[23] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals", *Sov. Phys. Dokl.*, vol. 10, pp. 707–710, 1966.

[24] S. Nagel. "News dataset available", Common Crawl, Oct. 2016, Accessed: Aug. 29, 2026. [Online]. Available: https://commoncrawl.org/blog/news-dataset-available

[25] Amazon Web Services. "What is Amazon S3?", Amazon Web Services, Accessed: Aug. 30, 2026. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

[26] G. Penedo, H. Kydlíček, A. Cappelli, M. Sasko, and T. Wolf, *DataTrove: Large scale data processing*, version 0.8.0, Jan. 19, 2026. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/huggingface/datatrove

[27] A. Barbaresi, "Trafilatura: A web scraping library and command-line tool for text discovery and extraction", in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, H. Ji, J. C. Park, and R. Xia, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 122–131. DOI: 10.18653/v1/2021.acl-demo.15

[28] A. Barbaresi, *Trafilatura*, version 1.11.0, Jun. 27, 2024. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/adbar/trafilatura/

[29] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification", in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds., Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431. [Online]. Available: https://aclanthology.org/E17-2068/

[30] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models", 2016. arXiv: 1612.03651.

[31] Ukrainian Drone Ecosystem Directory. "Search the directory", Ukrainian Drone Ecosystem Directory, Accessed: Aug. 29, 2026. [Online]. Available: https://www.drone-directory.com.ua/search-the-directory/

[32] Tech Force in UA. "Members of TFUA", Tech Force in UA, Accessed: Aug. 29, 2026. [Online]. Available: https://techforce.in.ua/en/members

[33] J. W. Rae et al., "Scaling language models: Methods, analysis & insights from training Gopher", 2021. arXiv: 2112.11446.

[34] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, *spaCy: Industrial-strength NLP*, version 3.8.11, Nov. 17, 2025. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/explosion/spaCy

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[35] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval", in *SIGIR '94*, B. W. Croft and C. J. van Rijsbergen, Eds., London: Springer London, 1994, pp. 232–241. DOI: 10.1007/978-1-4471-2099-5_24

[36] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-Embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation", in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 2318–2335. DOI: 10.18653/v1/2024.findings-acl.137

[37] X. H. Lù, "BM25S: Orders of magnitude faster lexical search via eager sparse scoring", 2024. arXiv: 2407.03618.

[38] M. F. Porter. "Snowball: A language for stemming algorithms", Snowball, Oct. 2001, Accessed: Aug. 29, 2026. [Online]. Available: https://snowballstem.org/texts/introduction.html

[39] R. Boulton, *PyStemmer*, version 3.0.0, May 8, 2025. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/snowballstem/pystemmer/

[40] D. Halaiko, *pymorphy3: Morphological analyzer for Russian and Ukrainian*, version 2.0.6, Oct. 9, 2025. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/no-plagiarism/pymorphy3

[41] Deutsche Welle. "Ukraine: Huge increase in drone production", Deutsche Welle, Nov. 18, 2023, Accessed: Aug. 29, 2026. [Online]. Available: https://www.dw.com/en/ukraine-huge-increase-in-drone-production/a-67476091

[42] P. Solin, *cleanco - clean organization names*, version 2.3, May 15, 2024. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/psolin/cleanco

[43] M. Bachmann, *RapidFuzz*, version 3.14.5, Apr. 7, 2026. DOI: 10.5281/zenodo.19450128

[44] P. Dyson and J. R. Coombs, *inflect*, version 5.6.2, Jul. 15, 2022. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/jaraco/inflect

[45] Neo4j, Inc., *Neo4j*, 2026. Accessed: Aug. 29, 2026. [Online]. Available: https://neo4j.com/

[46] Neo4j, Inc., *Neo4j bolt driver for Python*, version 6.1.0, Jan. 12, 2026. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/neo4j/neo4j-python-driver

[47] S. Upadhyay et al., "A large-scale study of relevance assessments with large language models using UMBRELA", in *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, ser. ICTIR '25, Padua, Italy: Association for Computing Machinery, 2025, pp. 358–368. DOI: 10.1145/3731120.3744605

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[48] C. Lawrence. “Quantum Systems and Frontline Robotics set up Europe’s first foreign drone production line for Ukraine”, Tech.eu, Dec. 15, 2025, Accessed: Aug. 29, 2026. [Online]. Available: https://tech.eu/2025/12/15/quantum-systems-and-frontline-robotics-set-up-europes-first-foreign-drone-production-line-for-ukraine/

[49] Ministry of Defence of Ukraine. “Ministry of Defence streamlines weapons codification and accelerates the supply of new developments to the military”, Ministry of Defence of Ukraine, Mar. 2, 2026, Accessed: Aug. 29, 2026. [Online]. Available: https://mod.gov.ua/en/news/ministry-of-defence-streamlines-weapons-codification-and-accelerates-the-supply-of-new-developments-to-the-military

[50] I. Kossov. “Ukraine will build 7 million drones in 2026—and it still won’t be enough”, Euromaidan Press, Jan. 26, 2026, Accessed: Aug. 29, 2026. [Online]. Available: https://euromaidanpress.com/2026/01/26/ukraine-aims-to-build-7-million-drones-in-2026-70-times-more-than-the-us/

[51] J. Thornhill. “Ukraine is winning the drone start-up war”, Financial Times, Apr. 10, 2025, Accessed: Aug. 29, 2026. [Online]. Available: https://www.ft.com/content/2c7d3c96-f6a8-4afa-bd88-5ea2fa39f5c1?syn-25a6b1a6=1

[52] O. Harmash, “Ukraine ramps up arms production, can produce 4 million drones a year, Zelenskiy says”, *Reuters*, Oct. 2, 2024. Accessed: Aug. 29, 2026. [Online]. Available: https://www.reuters.com/world/europe/ukraine-ramps-up-arms-production-can-produce-4-million-drones-year-zelenskiy-2024-10-02/

[53] Ministry of Defence of Ukraine. “A record 33,000 enemy UAVs were destroyed by interceptor drones in March—double the number recorded in the previous month”, Ministry of Defence of Ukraine, Apr. 8, 2026, Accessed: Aug. 29, 2026. [Online]. Available: https://mod.gov.ua/en/news/a-record-33-000-enemy-ua-vs-were-destroyed-by-interceptor-drones-in-march-double-the-number-recorded-in-the-previous-month

[54] M. Hunder, “Ukraine bets big on interceptor drones as low-cost air shield”, *Reuters*, Aug. 4, 2025. Accessed: Aug. 30, 2026. [Online]. Available: https://www.reuters.com/business/aerospace-defense/ukraine-bets-big-interceptor-drones-low-cost-air-shield-2025-08-04/

[55] O. Didur. “Безпілотний комплекс PD-1: народжений війною [The PD-1 unmanned aerial system: Born of war]”. in Ukrainian, АрміяInform [ArmyInform], Feb. 24, 2020, Accessed: Aug. 29, 2026. [Online]. Available: https://armyinform.com.ua/2020/02/24/bezpilotnyj-kompleks-pd-1-narodzhenyj-vijnoyu/

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[56] Army Technology. "PD-2 unmanned aerial system, Ukraine", Army Technology, Nov. 17, 2021, Accessed: Aug. 29, 2026. [Online]. Available: https://www.army-technology.com/projects/pd-2-unmanned-aerial-system-ukraine/

[57] R. Kotewa. "Crop protection application by drone: A Q&A with Kray Technologies CEO Dmytro Surdu", Global Ag Tech Initiative, Jan. 15, 2018, Accessed: Aug. 29, 2026. [Online]. Available: https://www.globalagtechinitiative.com/in-field-technologies/drones-uavs/crop-protection-application-by-drone-a-qa-with-kray-technologies-ceo-dmytro-surdu/

[58] Y. Pidhaina. "Галузі майбутнього: як безпілотники підкорюють Україну [Industries of the future: How drones are taking Ukraine by storm]". in Ukrainian, Mind.ua, Aug. 3, 2018, Accessed: Aug. 29, 2026. [Online]. Available: https://mind.ua/publications/20187343-galuzi-majbutnogo-yak-bezpilotniki-pidkoryuyut-ukrayinu

[59] U. Zaratiana, G. Pasternak, O. Boyd, G. Hurn-Maloney, and A. Lewis, "GLiNER2: Schema-driven multi-task learning for structured information extraction", in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, I. Habernal, P. Schulam, and J. Tiedemann, Eds., Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 130–140. DOI: 10.18653/v1/2025.emnlp-demos.10

[60] Boundary ML, *BAML: Basically a made-up language*, version 0.210.0, Oct. 1, 2025. Accessed: Aug. 29, 2026. [Online]. Available: https://github.com/BoundaryML/baml/releases/tag/0.210.0

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

APPENDIX A

Ukrainian Drone Manufacturers and Aliases

This appendix presents all seed nodes investigated in this thesis together with the names and aliases identified through manual research, as described in Section 3.2.

Table A.1: Ukrainian drone manufacturers and their aliases (English and Ukrainian) used as seed nodes.

Manufacturer	Aliases
"Meridian" n. S.P Korolyov	Meridian n. S.P.Korolyov, «Меридіан» ім. С. П.Корольова, «Меридіан» ім. С.П. Корольова
3DTech	*None*
Aerodrone	*None*
AERO ROZVIDKA	Аеророзвідка
Ailand Systems	Ailand
Air Combat Evolution	*None*
Airlogix	*None*
Antonov	Антонов
Athlon Avia	Атлон Авіа
AVATAR	*None*
BEZPEKA.PRO	BEZPEKA PRO, БЕЗПЕКА-ПРО
Blackbird	БЛЕКБЬОРД
Bluebird Tech	*None*
Buntar Aerospace	Buntar
Celebra	*None*
Culver Aerospace	Culver Aviation
DEF-C	*None*
Deviro	Девіро

Continued on the next page

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Manufacturer	Aliases
Drone Space Labs	*None*
dronebud	ЧДБ, Черкасидронбуд
eDrone	*None*
ELF Systems	*None*
Escadrone	Eskadron, Ескадрон
Falcons	*None*
Fire Point	Файєр Поінт
First Contact	*None*
FIRST PARSEC	*None*
FRDM	FRDM Group
Frontline Robotics	*None*
GENERAL CHERRY	Генерал Черешня
Gurzuf Defence	Gurzuf
huless	*None*
JET Company	J.E.T., jettech, Д.Ж.Е.Т.
Justifier Drones	*None*
Kray Technologies	Kray
Krylati	КРИЛАТІ
Luch Design Bureau	Design Bureau Luch, Engineering Bureau Luch, КБ Луч, Конструкторське бюро Луч, Луч КБ, Луч конструкторське бюро
Mara	MARA-DRONE, Mara Drone, МАРА
MaXon	MaXon Systems
MRIYA 4.5.0	МРІЯ 4.5.0
NEJET	Нежить
Omnitech	Omnitech Defense, ОмніТек, ОмніТек Дефенс
Pegasus Arms	Пегас Армс
Perun	*None*
PG Robotics	*None*
Phoenix drones	FX Drones
Piranha Tech	ПІРАНЬЯ ТЕХ
Raging Birds	ZLi Ptakhy, Злі Птахи
Ramzay	РАМЗАЙ
Reactive Drone	Reactive Drones, Реактивні Дрони
RIY Drone Project	RIY Drone
Roboneers	*None*
SKLBR	*None*
Sky Tactic	*None*
Skyassist	*None*
Skyeton	*None*
SkyFall	*None*
Smart UAV Technology	СМАРТ ЮЕВ ТЕХНОЛОДЖІС
Strategy Force Solution	Strategy Force Solutions
Svart Electronics Plus	Сварт Електронікс Плюс
TAF	TAF Drones, TAF Industries
TechEx	ТЕХ ЕКС, ТехЕкс

Continued on the next page

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Manufacturer	Aliases
Technotaras	Techno taras, Техно Тарас, Технотарас
Terminal Autonomy	ТЕРМІНАЛ АВТОНОМІ
The Fourth Law	TFL, Четвертий Закон
TORO	*None*
Trypillian	*None*
Twist Robotics	*None*
UA DYNAMICS	*None*
UKRJET	УКРДЖЕТ
Ukropchik	Calheon, Кальхеон, Укропчик
Ukrspecsystems	Укрспецсистемс
Ukrtekhno-Atom	Укртехно-Атом
United Unmanned Systems	UUS
Venator	Venator Technologies
VERBA	*None*
vidchayduhy	*None*
Vinnytsia Bees	Vin Bees, Vinnytski Bdzholy, Вінницькі бджоли
VYRIY	Viriy, Вирій
WarBirds	Бойові Птахи України
WeTrueGun	ВІТРУГАН
Wild Hornets	Dyki shershni, Дикі Шершні

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX B

Quality Filtering

This appendix describes how the Gopher quality filter used in Section 3.3 was tuned for the present use case using a human-annotated calibration dataset.

The Gopher quality filter consists of multiple subfilters. We dropped the `gopher_enough_stop_words` subfilter, which relies on a curated English stop-word list, from the original Gopher quality filter to keep the modified filter language-agnostic.

To construct the calibration dataset, we applied the default parameters listed in Table B.1 to a subsample of the entire corpus of extracted news articles before the seed-node filtering in Step 4. For English, Ukrainian, and German, we then sampled 35 articles that passed and 35 articles that were filtered out, so the dataset included likely low-quality texts. We included English and Ukrainian because they are directly relevant to the drone use case. German was included for possible later use cases. Both annotators annotated each of the resulting 210 articles using the annotation tool Prodigy [17]. A text was deemed to be of low quality if both annotators perceived it as such. Using this decision rule, we labelled 15 articles as low quality.

Using this dataset, we compared 41,472 parameter combinations in a grid search based on precision, recall, and the F_1 score. The complete parameter space is provided in Table B.1. With low quality as the positive class, we prioritised precision over recall, accepting that some low-quality texts would be retained to reduce the risk of excluding acceptable-quality texts containing supply-chain information. Several combinations achieved perfect precision with a recall of 0.2 and an F_1 score of 0.333, the highest recall observed among combinations with perfect precision. The selected parameter values and the reasons for their selection are shown in Table B.2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Parameter	Description	Tested values
min_doc_words	Minimum number of words a document needs to contain	10, 20, 30, 50 (default)
max_doc_words	Maximum number of words a document is allowed to contain	50,000, 100,000 (default)
min_avg_word_length	Minimum allowed average word length	1, 2, 3 (default)
max_avg_word_length	Maximum allowed average word length	10 (default), 12, 14, 16
max_symbol_word_ratio	Maximum allowed ratio of hash symbols or ellipses to words	0.05, 0.1 (default), 0.15, 0.2
max_bullet_lines_ratio	Maximum allowed fraction of lines that start with a bullet marker after leading whitespace	0.7, 0.85, 0.9 (default), 0.95
max_ellipsis_lines_ratio	Maximum allowed fraction of lines that end with an ellipsis	0.2, 0.3 (default), 0.4
min_non_alpha_words_ratio	Minimum required fraction of words containing at least one alphabetic character	0.5, 0.6, 0.65, 0.7, 0.75, 0.8 (default), 0.85, 0.9, 0.95

Table B.1: Parameter space used to tune the quality filter.

Parameter	Value	Reason for selection
min_doc_words	30	Highest value before precision decreased
max_doc_words	10,000	The tested values of 50,000 and 100,000 were both above the training set maximum of 4,002 words and therefore did not affect the evaluation. We selected 10,000 based on the domain knowledge that news articles are not expected to exceed this length
min_avg_word_length	3	Multiple values produced the same performance, so the Gopher value was retained
max_avg_word_length	10	Multiple values produced the same performance, so the Gopher value was retained
max_symbol_word_ratio	0.1	Multiple values produced the same performance, so the Gopher value was retained
max_bullet_lines_ratio	0.9	Multiple values produced the same performance, so the Gopher value was retained
max_ellipsis_lines_ratio	0.4	Only value available for the selected combination
min_non_alpha_words_ratio	0.6	Lowest value with the same performance, resulting in a less restrictive filter

Table B.2: Selected quality-filter parameters and reasons for their selection.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX C

Token-Level Seed-Node Matching

This appendix presents two fundamentally similar algorithms as pseudocode. Algorithm C.1 returns a Boolean flag indicating whether a seed-node match can be found in tokenised text. Algorithm C.2 extends this procedure by returning a token position for each occurrence of the seed node rather than a Boolean flag. Both algorithms use spaCy tokenisers with modified dash-splitting behaviour. The modifications split expressions at a dash when a number appears on either side rather than retaining them as a single token. Tokens are normalised using the same strategy as that described in Step 3 of Section 3.3.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Algorithm C.1: Token-level seed-node matching

Input: A seed node s, a document D, and its language l_D
Output: `true` if s has a valid token-level match in D

$T \leftarrow \text{tokenise}(\text{text}(D), l_D)$ `// Tokenised text`
$S \leftarrow$ empty sequence `// Seed-node tokens`
foreach $t \in \text{tokenise}(s, l_D)$ **do**
  $n \leftarrow \text{normalise}(t)$
  if n *is not empty* **then**
    append n to S
for $i \leftarrow 1$ **to** $|T| - |S| + 1$ **do**
  $C \leftarrow$ empty sequence `// Candidate tokens`
  $j \leftarrow i$
  if $\text{normalise}(T_i)$ *is empty* **then**
    continue `// Skip empty initial tokens`
  `/* Build a candidate of same length as seed node */`
  while $|C| < |S|$ *and* $j \leq |T|$ **do**
    $t \leftarrow T_j$
    $n \leftarrow \text{normalise}(t)$
    if t *is a line break, URL, or email address* **then**
      break
    if n *is not empty* **then**
      append n to C
    $j \leftarrow j + 1$
  if $C = S$ **then**
    return `true`
return `false`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithm C.2: Token-level seed-node matching with candidate-token position tracking

Input: A seed node s, a document D, and its language l_D
Output: $\mathcal{P}$, containing one representative token position for each valid match of s in D

$T \leftarrow \text{tokenise}(\text{text}(D), l_D)$ // Tokenised text
$S \leftarrow$ empty sequence // Seed-node tokens
$\mathcal{P} \leftarrow$ empty sequence // Positions of all matches
foreach $t \in \text{tokenise}(s, l_D)$ **do**
    $n \leftarrow \text{normalise}(t)$
    if n *is not empty* **then**
        append n to S
for $i \leftarrow 1$ **to** $|T| - |S| + 1$ **do**
    $C \leftarrow$ empty sequence // Candidate tokens
    $P \leftarrow$ empty sequence // Candidate token positions
    $j \leftarrow i$
    if $\text{normalise}(T_i)$ *is empty* **then**
        continue // Skip empty initial tokens
    /* Build a candidate of same length as seed node */
    while $|C| < |S|$ *and* $j \leq |T|$ **do**
        $t \leftarrow T_j$
        $n \leftarrow \text{normalise}(t)$
        if t *is a line break, URL, or email address* **then**
            break
        if n *is not empty* **then**
            append n to C
            append j to P // Position of token in text
        $j \leftarrow j + 1$
    if $C = S$ **then**
        append $\lceil \text{mean}(P) \rceil$ to $\mathcal{P}$ // Position of middle token
return $\mathcal{P}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX D

Deduplication Refinement

This appendix presents pseudocode for the deduplication refinement performed after candidate selection.

Algorithm D.1: Partitioning preliminary clusters by seed-node IDs

Input: Preliminary duplicate clusters $\mathcal{P}$ generated by DataTrove, where each $P = (r_P, C_P)$ consists of the representative r_P of P and the set C_P of duplicate candidates of P. All articles in P are written in the same language. Each representative and duplicate candidate is an article a with text, an article ID, a date, a token count, and a set I_a of associated seed-node IDs

Output: Seed-node-consistent clusters $\mathcal{Q}$, where each $Q \in \mathcal{Q}$ has the form $Q = (d_Q, A_Q)$ and consists of a cluster ID d_Q and the set A_Q of all articles assigned to that cluster

$\mathcal{Q} \leftarrow$ empty set `// Seed-node-consistent clusters`
foreach $P \in \mathcal{P}$ **do**
  $r_P \leftarrow$ representative of P
  $C_P \leftarrow$ duplicate candidates of P
  $A_P \leftarrow \{r_P\} \cup C_P$ `// All articles in` P
  $\Pi_P \leftarrow$ partition A_P by I_a, such that two articles belong to the same partition only if their seed-node ID sets are identical
  foreach *article set* $G \in \Pi_P$ **do**
    $d_G \leftarrow$ new cluster ID
    $Q_G \leftarrow (d_G, G)$
    add Q_G to $\mathcal{Q}$
return $\mathcal{Q}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Algorithm D.2: Selecting a representative by article length

Input: Seed-node-consistent clusters $\mathcal{Q}$, where each $Q \in \mathcal{Q}$ has the form $Q = (d_Q, A_Q)$ and consists of a cluster ID d_Q and an article set A_Q, and each article $a \in A_Q$ has text, an article ID, a date, a token count, and a set I_a of associated seed-node IDs

Output: Clusters $\mathcal{B}$, where each $B \in \mathcal{B}$ has the form $B = (d_B, r_B, C_B)$ and consists of a cluster ID d_B, a newly selected representative r_B, and a set C_B of duplicate candidates

$\mathcal{B} \leftarrow$ empty set `// Clusters with new representatives`
foreach $Q \in \mathcal{Q}$ **do**
  $d_Q \leftarrow$ cluster ID of Q
  $A_Q \leftarrow$ article set of Q
  if $|A_Q| < 10$ **then**
    $E_Q \leftarrow A_Q$ `// Eligible articles`
  else
    $Q_1, Q_3 \leftarrow$ first and third quartiles of the article token counts in A_Q
    $u \leftarrow Q_3 + 1.5(Q_3 - Q_1)$
    $E_Q \leftarrow$ articles in A_Q whose token count is at most u
  sort E_Q by decreasing token count, earlier date, and then by article ID
  $r_Q \leftarrow$ first article in E_Q
  $B_Q \leftarrow (d_Q, r_Q, A_Q \setminus \{r_Q\})$
  add B_Q to $\mathcal{B}$
return $\mathcal{B}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithm D.3: Validating candidate similarity to the representative

Input: Clusters $\mathcal{B}$, where each $B = (d_B, r_B, C_B)$ consists of a cluster ID d_B, a representative r_B, and duplicate candidates C_B. Each article $a \in \{r_B\} \cup C_B$ has text, an article ID, a date, a token count, and a set I_a of associated seed-node IDs

Output: Validated clusters $\mathcal{V}$, where each cluster has the form (d, r, D), with cluster ID d, representative r, and set D of duplicates, rejected candidate groups $\mathcal{F}$, where each $F \in \mathcal{F}$ is a sequence of candidates rejected from exactly one cluster B, and shingle mapping $\mathcal{H}$

$\tau \leftarrow 0.85$ `// Jaccard similarity threshold`
$\mathcal{V} \leftarrow$ empty set `// Validated clusters`
$\mathcal{F} \leftarrow$ empty set `// Rejected candidate groups`
$\mathcal{H} \leftarrow$ empty mapping `// Article-to-shingle mapping`
foreach $B \in \mathcal{B}$ **do** `// Generate shingles for each article`
  $r_B \leftarrow$ representative of B
  $C_B \leftarrow$ duplicate candidates of B
  foreach $a \in \{r_B\} \cup C_B$ **do**
    $\mathcal{H}[a] \leftarrow \text{shingle}(a, 5)$ `// DataTrove 5-token shingling`
foreach $B \in \mathcal{B}$ **do**
  $d_B \leftarrow$ cluster ID of B
  $r_B \leftarrow$ representative of B
  $C_B \leftarrow$ duplicate candidates of B
  $V_B \leftarrow$ empty set `// Validated candidates`
  $F_B \leftarrow$ empty sequence `// Rejected candidates`
  foreach $c \in C_B$ **do**
    $j \leftarrow \text{jaccard}(\mathcal{H}[r_B], \mathcal{H}[c])$ `// Jaccard similarity`
    if $j \geq \tau$ **then**
      add c to V_B
    else
      append c to F_B
  if V_B *is not empty* **then**
    add (d_B, r_B, V_B) to $\mathcal{V}$
  else
    add $(-1, r_B, \{\})$ to $\mathcal{V}$ `// −1 represents a singleton`
  if F_B *is not empty* **then**
    add F_B to $\mathcal{F}$
return $\mathcal{V}, \mathcal{F}, \mathcal{H}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithm D.4: Reclustering rejected candidates and combining the results

Input: Validated clusters $\mathcal{V}$, a set $\mathcal{S}$ of articles that were never assigned to a cluster by DataTrove, rejected candidate groups $\mathcal{F}$, and shingle mapping $\mathcal{H}$. Each article a has text, an article ID, a date, a token count, and a set I_a of associated seed-node IDs

Output: Final clusters $\mathcal{R}$, where each cluster has the form (d, r, D)

$\tau \leftarrow 0.85$ `// Jaccard similarity threshold`
$\mathcal{R} \leftarrow \mathcal{V}$
foreach *article* $a \in \mathcal{S}$ *never assigned to a cluster by DataTrove* **do**
  add $(-1, a, \{\})$ to $\mathcal{R}$ `// −1 represents a singleton`
foreach $F \in \mathcal{F}$ **do**
  sort F by decreasing token count and then by article ID
  $A \leftarrow$ empty set `// Assigned articles`
  foreach $r \in F$ **do** `// Candidate representative`
    if $r \in A$ **then**
      continue
    $D_r \leftarrow$ empty set `// Duplicates`
    add r to A
    foreach c *after* r *in* F **do**
      if $c \notin A$ *and* $\text{jaccard}(\mathcal{H}[r], \mathcal{H}[c]) \geq \tau$ **then**
        add c to D_r
        add c to A
    if D_r *is not empty* **then**
      $d_r \leftarrow$ new cluster ID
      add (d_r, r, D_r) to $\mathcal{R}$
    else
      add $(-1, r, \{\})$ to $\mathcal{R}$ `// −1 represents a singleton`
return $\mathcal{R}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX E

SilverExtension Sampling

This appendix presents pseudocode for sampling the article–seed-node pairs in SilverExtension.

Algorithm E.1: SilverExtension sampling for one language

Input: SilverBase $\mathcal{S}_b$ and ranking collections $\mathcal{R} = \{\mathcal{R}_d, \mathcal{R}_s, \mathcal{R}_h\}$, where $\mathcal{R}_d$, $\mathcal{R}_s$, and $\mathcal{R}_h$ contain the rankings produced by the dense, sparse, and hybrid retrieval configurations retained for pooling, respectively. Within each collection, the rankings are ordered by decreasing mean Recall@10,000

Output: SilverExtension $\mathcal{S}_e$

foreach $\mathcal{R}_m \in \mathcal{R}$ **do**
  $\mathcal{U} \leftarrow \varnothing$ `// Unique candidates`
  $c(x) \leftarrow 0$ for every candidate x `// Candidate occurrence counts`
  foreach $r_i \in \mathcal{R}_m$ **do**
    foreach $x \in r_i[:10{,}000]$ **do**
      $\mathcal{U} \leftarrow \mathcal{U} \cup \{x\}$
      $c(x) \leftarrow c(x) + 1$
    if $|\mathcal{U}| > 20{,}000$ **then**
      break
  foreach $x \in \mathcal{U}$ **do**
    $p(x) \leftarrow \dfrac{c(x)}{\sum_{y \in \mathcal{U}} c(y)}$
  $\mathcal{X}_m \leftarrow \text{weightedSample}(\mathcal{U}, 10{,}000, p)$ `// Without replacement`
$\mathcal{S}_e \leftarrow (\mathcal{X}_d \cup \mathcal{X}_s \cup \mathcal{X}_h) \setminus \mathcal{S}_b$
return $\mathcal{S}_e$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX F

Additional Corpus Composition Plots

This appendix contains additional plots that provide insight into the composition of the preprocessed and deduplicated CC-News corpus.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

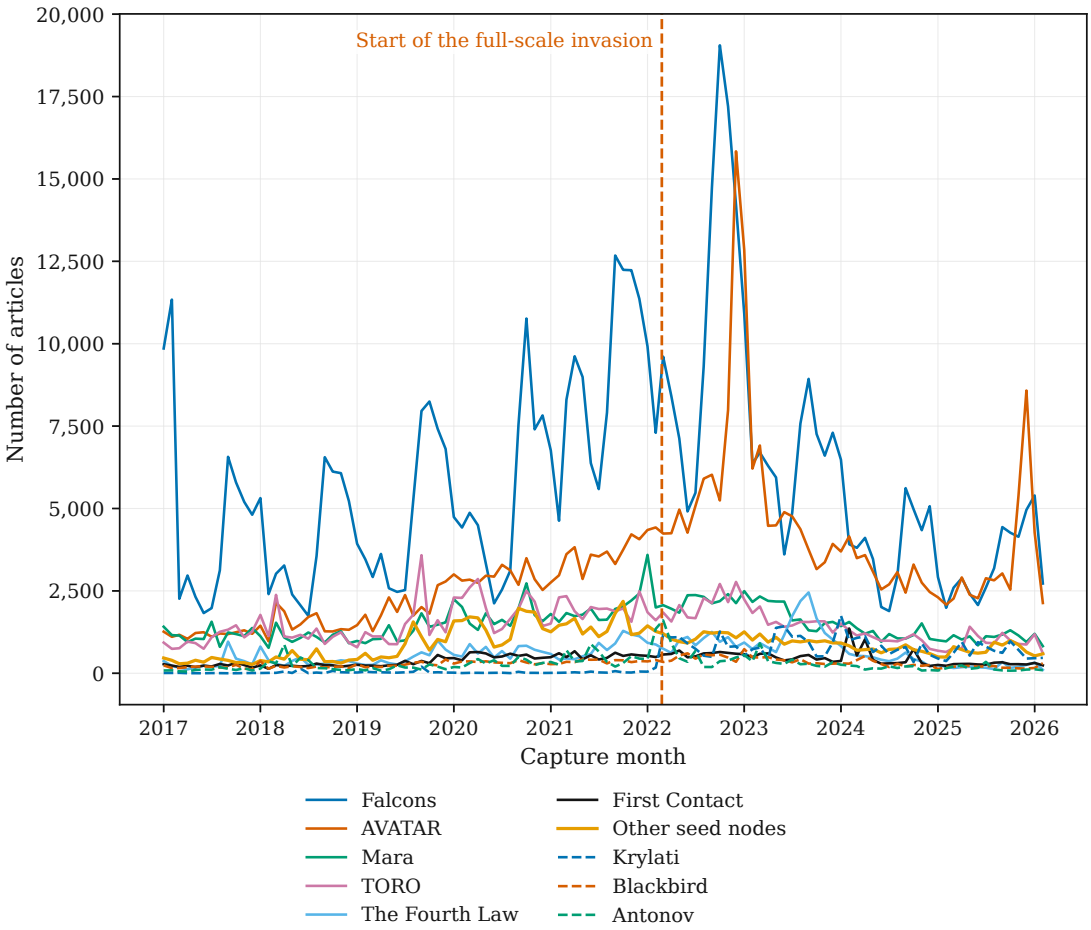

Figure F.1: Number of retained articles by capture month for each of the nine seed nodes with the largest total number of associated articles. All remaining seed nodes are aggregated as Other seed nodes. An article associated with several seed nodes contributes to each relevant series. The dashed line marks the start of the full-scale invasion on 24 February 2022.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

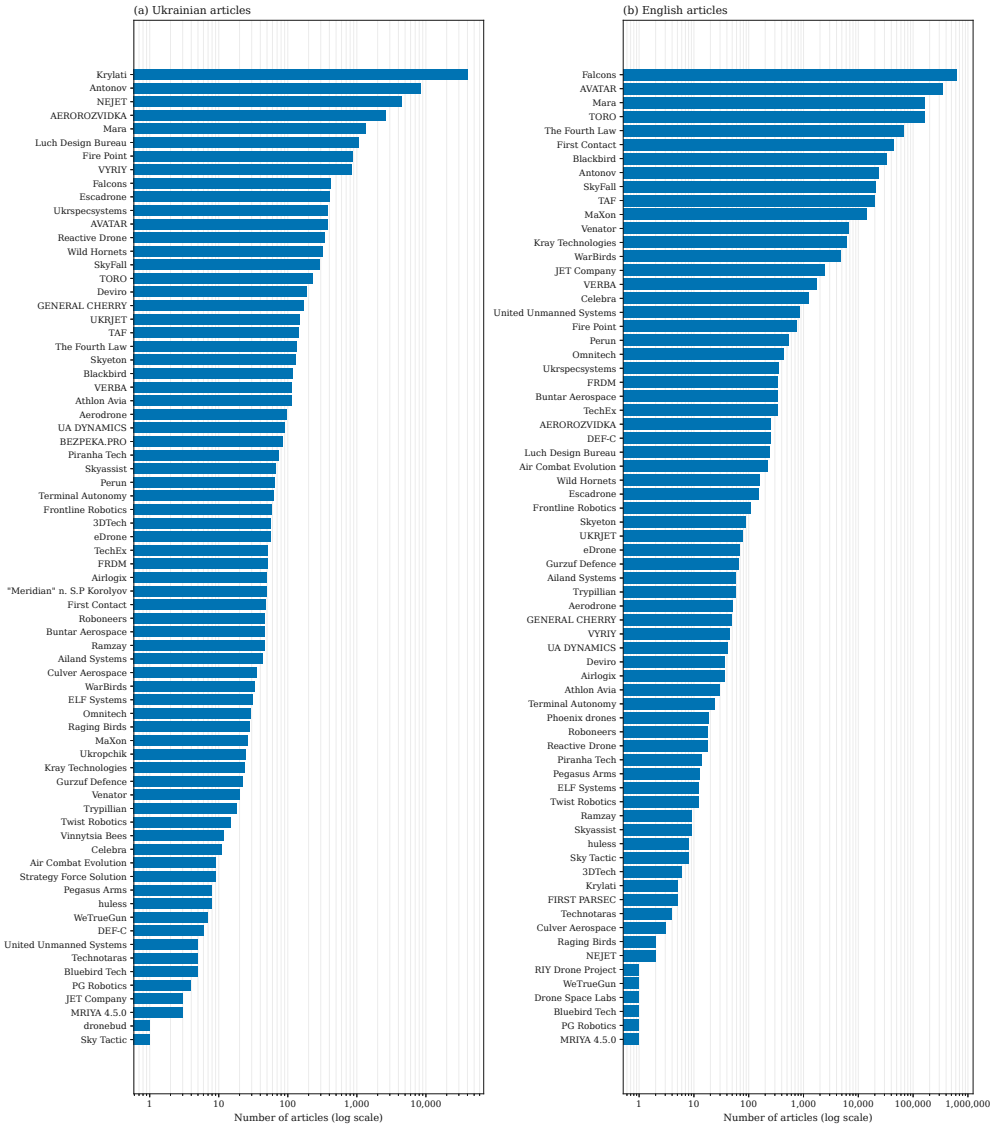

Figure F.2: Number of retained articles associated with each seed node after deduplication.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX G

Detailed Named Entity Recognition Results

This appendix provides a more detailed breakdown of the NER evaluation results presented in Section 6.4.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

		Human-selected input			Pipeline-generated input		
Source language	Entity type	Precision	Recall	F_1	Precision	Recall	F_1
English	DATE	0.867 ± 0.000	1.000 ± 0.000	0.929 ± 0.000	0.631 ± 0.050	1.000 ± 0.000	0.773 ± 0.037
	PRODUCT_TYPE	0.924 ± 0.014	0.765 ± 0.020	0.837 ± 0.014	0.844 ± 0.034	0.722 ± 0.015	0.778 ± 0.018
	NAMED_PRODUCT	0.957 ± 0.018	0.934 ± 0.016	0.945 ± 0.016	0.929 ± 0.019	0.919 ± 0.034	0.924 ± 0.023
	GPE	0.935 ± 0.020	0.902 ± 0.017	0.918 ± 0.016	0.825 ± 0.037	0.880 ± 0.022	0.851 ± 0.022
	ORG	0.942 ± 0.007	0.950 ± 0.012	0.946 ± 0.007	0.915 ± 0.014	0.952 ± 0.009	0.933 ± 0.011
	Overall	0.935 ± 0.007	0.877 ± 0.010	0.905 ± 0.006	0.871 ± 0.020	0.858 ± 0.007	0.864 ± 0.010
Ukrainian	DATE	0.857 ± 0.040	0.900 ± 0.044	0.878 ± 0.040	0.677 ± 0.052	0.925 ± 0.026	0.781 ± 0.036
	PRODUCT_TYPE	0.913 ± 0.011	0.821 ± 0.011	0.864 ± 0.008	0.776 ± 0.021	0.794 ± 0.023	0.785 ± 0.019
	NAMED_PRODUCT	0.941 ± 0.017	0.880 ± 0.015	0.909 ± 0.015	0.819 ± 0.017	0.843 ± 0.020	0.831 ± 0.014
	GPE	0.970 ± 0.008	0.940 ± 0.017	0.954 ± 0.008	0.750 ± 0.025	0.778 ± 0.034	0.764 ± 0.027
	ORG	0.855 ± 0.011	0.917 ± 0.014	0.885 ± 0.011	0.778 ± 0.034	0.907 ± 0.012	0.837 ± 0.022
	Overall	0.906 ± 0.006	0.867 ± 0.006	0.886 ± 0.005	0.777 ± 0.022	0.831 ± 0.017	0.803 ± 0.017
Combined	DATE	0.862 ± 0.021	0.945 ± 0.024	0.901 ± 0.022	0.652 ± 0.025	0.959 ± 0.015	0.776 ± 0.019
	PRODUCT_TYPE	0.917 ± 0.008	0.799 ± 0.008	0.854 ± 0.005	0.800 ± 0.017	0.765 ± 0.015	0.782 ± 0.012
	NAMED_PRODUCT	0.947 ± 0.010	0.901 ± 0.009	0.923 ± 0.008	0.860 ± 0.008	0.872 ± 0.020	0.866 ± 0.012
	GPE	0.952 ± 0.011	0.920 ± 0.008	0.935 ± 0.008	0.790 ± 0.028	0.831 ± 0.019	0.810 ± 0.020
	ORG	0.904 ± 0.007	0.936 ± 0.008	0.920 ± 0.006	0.853 ± 0.019	0.933 ± 0.009	0.891 ± 0.013
	Overall	0.919 ± 0.005	0.872 ± 0.005	0.895 ± 0.003	0.819 ± 0.015	0.844 ± 0.009	0.831 ± 0.010

Table G.1: Named entity recognition evaluation results by article source language and entity type, including results for both languages combined, for the human-selected and pipeline-generated inputs. Overall values are micro-averaged. All values are reported as mean ± sample standard deviation across ten runs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX H

Detailed Relationship Extraction Results

This appendix provides a more detailed breakdown of the RE evaluation results presented in Section 6.5.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Source language	Relationship type	Relationship variant	Human-selected input			Pipeline-generated input		
			Precision	Recall	F_1	Precision	Recall	F_1
English	is_type	Core	0.751 ± 0.037	0.845 ± 0.040	0.795 ± 0.035	0.670 ± 0.053	0.705 ± 0.024	0.686 ± 0.032
	located_in	Core	0.897 ± 0.045	0.736 ± 0.035	0.808 ± 0.032	0.781 ± 0.040	0.729 ± 0.056	0.752 ± 0.032
		Full	0.844 ± 0.050	0.693 ± 0.048	0.761 ± 0.044	0.728 ± 0.060	0.679 ± 0.061	0.701 ± 0.047
	makes	Core	0.688 ± 0.041	0.887 ± 0.022	0.774 ± 0.026	0.658 ± 0.038	0.769 ± 0.027	0.709 ± 0.031
		Full	0.670 ± 0.046	0.863 ± 0.025	0.753 ± 0.033	0.640 ± 0.041	0.747 ± 0.026	0.689 ± 0.033
	supplies_to	Core	0.567 ± 0.161	0.750 ± 0.264	0.613 ± 0.106	0.543 ± 0.272	0.700 ± 0.258	0.596 ± 0.252
		Full	0.283 ± 0.300	0.350 ± 0.242	0.290 ± 0.226	0.203 ± 0.331	0.250 ± 0.354	0.219 ± 0.336
	Overall	Core	0.716 ± 0.028	0.858 ± 0.017	0.780 ± 0.017	0.668 ± 0.040	0.746 ± 0.021	0.705 ± 0.029
		Full	0.695 ± 0.030	0.833 ± 0.017	0.757 ± 0.020	0.646 ± 0.042	0.721 ± 0.023	0.681 ± 0.032
Ukrainian	is_type	Core	0.713 ± 0.033	0.826 ± 0.023	0.765 ± 0.021	0.653 ± 0.035	0.773 ± 0.023	0.707 ± 0.022
	located_in	Core	0.871 ± 0.021	0.789 ± 0.051	0.827 ± 0.036	0.817 ± 0.022	0.567 ± 0.063	0.667 ± 0.044
		Full	0.871 ± 0.021	0.789 ± 0.051	0.827 ± 0.036	0.817 ± 0.022	0.567 ± 0.063	0.667 ± 0.044
	makes	Core	0.753 ± 0.017	0.873 ± 0.015	0.808 ± 0.012	0.679 ± 0.035	0.776 ± 0.027	0.724 ± 0.030
		Full	0.698 ± 0.023	0.810 ± 0.025	0.750 ± 0.022	0.626 ± 0.034	0.715 ± 0.026	0.667 ± 0.029
	supplies_to	Core	0.256 ± 0.049	0.325 ± 0.065	0.285 ± 0.054	0.293 ± 0.064	0.325 ± 0.065	0.307 ± 0.061
		Full	0.127 ± 0.043	0.163 ± 0.060	0.142 ± 0.050	0.126 ± 0.046	0.138 ± 0.040	0.131 ± 0.043
	Overall	Core	0.727 ± 0.015	0.831 ± 0.011	0.775 ± 0.008	0.663 ± 0.028	0.739 ± 0.019	0.699 ± 0.021
		Full	0.690 ± 0.013	0.788 ± 0.014	0.735 ± 0.009	0.625 ± 0.028	0.697 ± 0.020	0.659 ± 0.022
Combined	is_type	Core	0.727 ± 0.030	0.833 ± 0.018	0.776 ± 0.021	0.658 ± 0.032	0.747 ± 0.018	0.699 ± 0.021
	located_in	Core	0.882 ± 0.022	0.766 ± 0.027	0.819 ± 0.021	0.798 ± 0.029	0.637 ± 0.042	0.708 ± 0.027
		Full	0.860 ± 0.025	0.747 ± 0.031	0.799 ± 0.026	0.771 ± 0.037	0.616 ± 0.036	0.684 ± 0.025
	makes	Core	0.724 ± 0.017	0.879 ± 0.010	0.794 ± 0.008	0.670 ± 0.024	0.773 ± 0.021	0.718 ± 0.021
		Full	0.685 ± 0.014	0.832 ± 0.017	0.751 ± 0.009	0.631 ± 0.025	0.728 ± 0.020	0.676 ± 0.021
	supplies_to	Core	0.313 ± 0.052	0.410 ± 0.099	0.354 ± 0.070	0.343 ± 0.082	0.400 ± 0.067	0.367 ± 0.071
		Full	0.153 ± 0.056	0.200 ± 0.082	0.173 ± 0.066	0.142 ± 0.090	0.160 ± 0.084	0.150 ± 0.087
	Overall	Core	0.722 ± 0.017	0.841 ± 0.010	0.777 ± 0.008	0.664 ± 0.023	0.742 ± 0.016	0.701 ± 0.018
		Full	0.692 ± 0.014	0.806 ± 0.010	0.744 ± 0.006	0.633 ± 0.024	0.707 ± 0.017	0.668 ± 0.019

Table H.1: Relationship-extraction evaluation results by article source language, relationship type, and relationship variant for the human-selected and pipeline-generated inputs, including results for both languages combined. All values are reported as mean ± sample standard deviation across ten runs. The full relationship variant is not applicable to `is_type`.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

APPENDIX I

Prompt Templates

This appendix presents the rendered system prompts and example user inputs. All prompts were modelled using BAML [60].

I.1 Keyword Query Group Translation

Keyword group	Description
Country	Keywords for mentions of Ukraine or descriptions of Ukrainian entities.
Domain-specific	Keywords for drone- and UAV-related texts.
Organisation	Keywords for texts about organisations or companies.
Manufacturing	Keywords for texts about manufacturing and production.
Supply relationship	Keywords for texts about supply and purchasing relationships.
Affiliation	Keywords for texts about people connected to companies or organisations.
Company location	Keywords for texts about company and organisation locations.
Part of	Keywords for texts about part-of or component relationships between products.

Table I.1: Descriptions of the keyword groups provided as user input when translating English keywords into Ukrainian.

I.1.1 System Prompt

```
You are translating English search keywords into single-word BM25 query terms in the target
↪  language: Ukrainian.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Context:
The queries are for finding Ukrainian-language information about UAV/drone companies,
↪  manufacturers, suppliers, products, affiliated persons, locations, facilities, and
↪  components.


Task:
You are given a list of English keywords. Return single-word search terms that are safe to use
↪  in BM25 retrieval.
Use the provided keyword description to identify the intended semantic category of the English
↪  keywords.

Rules:
- Return one JSON entry for each English input term.
- keep must contain only high-confidence single-word terms.
- avoid must contain wrong-sense, broad, uncertain, or unusable terms.
- Multi-word translations are forbidden in keep.
- Do not replace a correct multi-word translation with a noisy single-word approximation.
- A single word may be a normal word, abbreviation, acronym, or hyphenated compound.
- Hyphenated compounds count as single words only if they are normally written and treated as
↪  one unit in Ukrainian.
- Abbreviations count as single words.
- English acronyms may be kept only if they are commonly used in Ukrainian texts for this
↪  context.
- If there is no safe single-word equivalent, return an empty keep list.
- Do not invent rare or unnatural terms.
- Because the retrieval pipeline normalizes or stems terms later, return normal word forms
↪  rather than stems, truncations, or artificial forms.
- Include only one representative normal form per stem or normalized form, and omit variants
↪  that would normalize alike.
- Prefer terms likely to appear in sources relevant to the provided context.

Answer with a JSON Array using this schema:
[
  {
    // The original English term to be translated.
    english_term: string,
    // Single-token search terms in the target language that clearly fit the context and should
    ↪  be used.
    // Include only high-confidence terms.
    keep: string[],
    // Wrong-sense, misleading, multi-word, rare, unnatural, or uncertain translations.
    // If unsure, put the term here rather than in keep.
    avoid: string[],
    // A brief English explanation of the decision, especially for avoided terms.
    reason: string,
  }
]
```

I.1.2 Example User Prompt

```
Keywords: ["interceptor", "reconnaissance", "surveillance", "strike", "payload", "kamikaze",
↪  "bomber", "loitering", "drone", "aerial", "unmanned", "FPV", "UAV", "UAS", "quadcopter"]
Keyword description "Keywords for drone- and UAV-related texts."
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

I.2 Natural-Language Query Generation

I.2.1 Single-Query Generation

System Prompt

```
You are an expert query formulator specializing in dense semantic retrieval.

Your task is to transform user-provided keyword strings into one natural-language query for
↪  dense semantic retrieval. The query will be embedded and used to rank text by cosine
↪  similarity.

Input interpretation:
- Each item in keyword_strings is a required semantic component of the search intent.
- Each item may contain one keyword or a space-separated group of related keywords.
- The keywords indicate meaning; they are not mandatory output terms.

Query construction:
- Represent the semantic content of every item in the keyword strings in the final query.
- Do not mechanically copy or include every keyword. Use natural wording that captures the
↪  meaning more clearly when possible.
- Preserve the overall search intent.
- Write like a relevant document passage, not a Boolean search query, command, keyword list, or
↪  imperative request such as "Find..." or "Search for...".

Output requirements:
- Return exactly one query string.
- Do not return markdown, bullets, explanations, labels, or quotation marks around the query.
- Use the same primary language as the supplied keyword strings.
- Use a maximum of 3 sentences.
```

Example User Prompt

```
Keyword strings:
["Ukraine Ukrainian", "interceptor reconnaissance surveillance strike payload kamikaze bomber
↪  loitering drone aerial unmanned FPV UAV UAS quadcopter"]
```

I.2.2 Multiple-Query Generation

System Prompt

```
You are an expert query formulator specializing in dense semantic retrieval.

Your task is to transform user-provided keyword strings into multiple natural-language queries
↪  for dense semantic retrieval. Each query will be embedded and used independently to rank
↪  text by cosine similarity.

Input interpretation:
- Each item in keyword_strings is a required semantic component of the search intent.
- Each item may contain one keyword or a space-separated group of related keywords.
- The keywords indicate meaning; they are not mandatory output terms.

Query construction:
- Represent the semantic content of every item in the keyword strings in the final query.
- Do not mechanically copy or include every keyword. Use natural wording that captures the
↪  meaning more clearly when possible.
- Preserve the overall search intent.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
- Write like a relevant document passage, not a Boolean search query, command, keyword list, or
↪  imperative request such as "Find..." or "Search for...".

Multi-query diversity:
- The queries must be fundamentally different from each other while covering the same core
↪  semantics.
- Do not produce minor paraphrases, reordered wording, or near-duplicates.
- Each query should approach the same underlying information need from a genuinely different
↪  angle.

Output requirements:
- Return exactly 4 query strings.
- Return each query as a new list entry
- Do not return markdown, bullets, explanations, labels, or quotation marks around the queries.
- Use the same primary language as the supplied keyword strings.
- Each query must use a maximum of 3 sentences.
- The queries must be meaningfully distinct from each other while still covering the same core
↪  semantics.


Answer with a JSON Array using this schema:
string[]
```

Example User Prompt

```
Number of queries:
4

Keyword strings:
["Ukraine Ukrainian", "interceptor reconnaissance surveillance strike payload kamikaze bomber
↪  loitering drone aerial unmanned FPV UAV UAS quadcopter"]
```

I.3 Relevance Classification

I.3.1 System Prompt

```
You are an expert in identifying entity mentions in a given text.

The user will provide you with a text passage, the lowercase entity name of the target entity,
↪  optional lowercase aliases of the entity, an entity description, and a business activity.
↪  Your goal is to decide whether the text clearly mentions this specific target entity and
↪  clearly links the target entity to its specified business activity. Decide only based on
↪  the given information.

Your decision must be either TRUE or FALSE.

Decision rules:

- Return TRUE only when the text explicitly mentions the target entity, the surrounding wording
↪  clearly identifies the target entity rather than a different entity with the same or a
↪  similar name, and the text clearly shows that the target entity itself is active in the
↪  specified business activity.

- Return FALSE when the text does not refer to the target entity. For example, this applies if a
↪  mention refers to a different entity with the same or a similar name, or if the text is
↪  about something entirely different.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
- Return FALSE when the text refers to the target entity but does not provide clear evidence
↪  that the entity itself is active in the specified business activity. For example, if an
↪  entity produces product A and product B, and the provided business activity field is about
↪  product A, return FALSE if the provided text only discusses that entity in the context of
↪  product B.

- If the evidence is ambiguous as to whether the phrase refers to the target entity, or whether
↪  the text links the target entity to its specified business activity, return FALSE.

Additional instructions:
- Do not rely on exact string overlap alone! Also consider the context, because the same phrase
↪  can refer to the target entity, another entity of the same or a different type, or just be a
↪  random string match.

- Names and aliases have been normalized only by lowercasing. Spaces, hyphens, quotation marks,
↪  apostrophes, and other separators are still meaningful and have been preserved. For
↪  example, acme-labs may appear as Acme-Labs. If no aliases are provided, assume that there
↪  are no known aliases.

### Examples

Example 1
Entity name: Sony
Aliases: []
Entity description: Sony is a Japanese entity that manufactures, produces, develops, or designs
↪  cameras, mobile phones, computer games, and consoles.
Business activity: Production, development, manufacturing, or design of cameras and vision
↪  equipment.
Text: Sony is producing the game console PlayStation 5.
Expected decision: FALSE
Why: The text mentions Sony, but only as a game console producer. It does not link Sony to its
↪  camera-related business activity.

Example 2
Entity name: Tesla
Aliases: []
Entity description: Tesla is an American entity that manufactures, produces, develops, or
↪  designs electric vehicles, vehicle batteries, charging equipment, solar energy products,
↪  energy storage products, autonomous driving software, AI hardware, and robots.
Business activity: Production, development, manufacturing, or design of robots and robotics
↪  equipment.
Text: Tesla announced a new electric car at its annual product event.
Expected decision: FALSE
Why: The text identifies Tesla in a car-related context, but does not link Tesla to its
↪  robot-related business activity.

Example 3
Entity name: Toyota
Aliases: []
Entity description: Toyota is a Japanese entity that manufactures, produces, develops, or
↪  designs passenger and commercial vehicles, electrified and hydrogen powertrains, vehicle
↪  batteries, autonomous and connected mobility software, mobility services, motorsports
↪  vehicles, lunar rovers, and robots.
Business activity: Production, development, manufacturing, or design of electric vehicles.
Text: Toyota manufactures the fully electric bZ4X and opened a new battery testing facility.
Expected decision: TRUE
Why: The text identifies Toyota and clearly links it to electric-vehicle manufacturing.


Example 4
Entity name: Ferrari
Aliases: []
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Entity description: Ferrari is an Italian entity that manufactures, produces, develops, or
↪  designs road cars, sports cars, supercars, and Formula 1 racing vehicles.
Business activity: Production, development, manufacturing, or design of road cars.
Text: Ferrari unveiled the SF-23, its Formula 1 racing car for the new season.
Expected decision: FALSE
Why: The text clearly links Ferrari to a Formula 1 racing car, which is adjacent to road cars
↪  but is a different product category. Because the requested business activity is road cars,
↪  the text does not show Ferrari acting in that specific activity.

Answer in JSON using this schema:
{
  // Return your final decision here: TRUE or FALSE.
  relevant: bool,
  // Briefly justify your decision in a maximum of two concise sentences.
  reason: string,
}
```

I.3.2 Example User Prompt

```
Entity name: antonov
Known aliases: ["антонов"]
Entity description: Antonov is a Ukrainian entity (e.g., company, volunteer group, NGO,
↪  organization) that manufactures, produces, develops, or designs drones.
Business activity: Production, development, manufacturing, or design of drones and unmanned
↪  aerial vehicles.
Text: The Ukrainian aircraft manufacturer Antonov expanded into drone production
It is planning to build reconnaissance drones
See also: Zelensky does a powerful speech
```

I.4 Relevant Segment Selection

I.4.1 System Prompt

```
You are an expert annotator for segment-level relevance judgment.

You are given a text split into numbered segments, a target entity, and optional aliases for
↪  the target entity.

Your task is to identify which segments are relevant to the target entity based on the concept
↪  definitions and segment relevance rules below.

Use only the provided text, target entity, aliases, and the definitions and rules below. Do not
↪  use outside knowledge.

In `relevant_segments`, return only segment IDs that are explicitly present in the input text.
↪  Do not invent segment IDs.

Use only the numeric IDs from the `<Segment {id}>` markers. Return the IDs as strings, for
↪  example `["1", "4"]`.


## Concept Definitions

Below are key concept definitions used by the relevance rules.

### 1. Target entity and aliases
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
The **target entity** is the company, volunteer group, NGO, or organization the task is about.

**Aliases** are alternative names, spellings, or abbreviations for the target entity. They
↪  include the provided aliases and any additional alias that the article itself clearly
↪  establishes as referring to the target entity.

Names and aliases have been normalized only by lowercasing. Spaces, hyphens, quotation marks,
↪  apostrophes, and other separators are still meaningful and have been preserved. For
↪  example, beta-industries may appear as Beta-Industries.

Only use an article-established alias when it is clear from the surrounding context that the
↪  alias refers to the target entity. Do not infer aliases from outside knowledge.

Example: If the target entity is `Alpha Systems` and the provided alias is `A-Systems`, both
↪  count as target-entity mentions when the context clearly refers to this entity.

### 2. Product reference

A **product reference** is a mention of a product, service, system, platform, component,
↪  software, or technical offering.

Product references include **model names / named products** and **product types**.

A **model name / named product** refers to a particular product offered by a particular
↪  company. It usually identifies a specific model, version, or product line with particular
↪  features or specifications.

A **product type** refers to a generic product, component, system, software, or technology
↪  category shared by multiple similar products. A product type is not tied to one specific
↪  model, version, or product line from one company.


### Examples

- `Tesla Model X`
  - company: `Tesla`
  - model name / named product: `Model X`
  - product type: `electric vehicle`

- `Sony Alpha DSLR-A100`
  - company: `Sony`
  - model name / named product: `Alpha DSLR-A100`
  - product type: `digital single-lens reflex camera`

- `Apple iPhone 15`
  - company: `Apple`
  - model name / named product: `iPhone 15`
  - product type: `smartphone`

- `Bosch Rexroth IndraDrive`
  - company: `Bosch Rexroth`
  - model name / named product: `IndraDrive`
  - product type: `servo drive system`

### 3. Target-entity product

A **target-entity product** is a product reference that the article clearly establishes as
↪  something the target entity makes, develops, designs, sells, supplies, or otherwise offers.

A broad product category is not a target-entity product merely because it is mentioned together
↪  with the target entity. For example, `electric vehicles`, `sensors`, or `cameras` are not
↪  target-entity products unless the article clearly establishes that the target entity offers
↪  products of that type.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

````
Identifying whether a product type or model name / named product is a **target-entity product**
↪  is part of the annotation task.


Example 1:

```text
Alpha Systems manufactures the AX-10 industrial camera.
```

Target entity: `Alpha Systems`

Inferred target-entity product: `AX-10 industrial camera`

Here, the text clearly establishes that Alpha Systems manufactures the AX-10 industrial camera.

Example 2:

```text
Alpha Systems presented its new industrial camera.
```

Target entity: `Alpha Systems`

Inferred target-entity product: `industrial camera`

Here, the text clearly establishes that Alpha Systems offers the industrial camera.

### 4. Contextual reference

A **contextual reference** is a phrase that does not fully name an entity, product, person,
↪  organization, or place, but clearly refers to it through the surrounding article context.

The necessary context can come from one or more connected segments. These segments do not need
↪  to be consecutive, but the reference must still be clearly established.

To be relevant, a **contextual reference** must refer to a specific **target entity** or
↪  **target-entity product** that is clearly established in the article context.

### Examples

`Alpha Systems manufactures industrial cameras. These devices are used in factories.`

Here, `These devices` refers to Alpha Systems' industrial cameras.

`Alpha Systems produces the industrial camera AX-10. The camera is used to optimize factories.`

Here, `the camera` refers to the inferred target-entity product.

`Alpha Systems produces the AX-10 industrial camera. Industrial cameras are used more often in
↪  factories.`

Here, `Industrial cameras` in the second sentence does not refer to the `AX-10 industrial
↪  camera` by Alpha Systems. It is a generic product-category statement.

### 5. Clear attribution

**Clear attribution** means that the text clearly connects a fact, property, product, person,
↪  organization, place, or relationship to a specific entity, product, person, organization,
↪  or place.
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

````
Clear attribution can come from the same segment or from multiple other segments that provide
↪  the necessary context.

### Examples

```text
<Segment 1>
Alpha Systems produces industrial cameras.

<Segment 2>
The company is based in Munich.

<Segment 3>
Its cameras use thermal sensors.

<Segment 4>
Alpha Systems supplies cameras to Beta Corp.
```

Target entity: `Alpha Systems`

In this example, the location (`<Segment 2>`), thermal sensors (`<Segment 3>`), and the supply
↪  relationship (`<Segment 4>`) are clearly attributed to `Alpha Systems` or its cameras.

If an attribution is vague or unclear, it is not a clear attribution.

Example of no clear attribution:

```text
<Segment 1>
Alpha Systems produces industrial cameras.

<Segment 2>
Thermal sensors are often used in industrial cameras.
```

Target entity: `Alpha Systems`

Segment 2 is a generic category statement (see definition 7) and does not refer to `Alpha
↪  Systems`, so we cannot attribute thermal sensors to `Alpha Systems` or its cameras.

### 6. Minimal context chain

A **minimal context chain** is the smallest connected set of segments that makes it clear how a
↪  fact, product, role, or relationship is linked to the **target entity** or a
↪  **target-entity product**.

The segments in a minimal context chain do not need to be consecutive, but they must form a
↪  clearly connected logical chain. Distant segments are part of the same chain only when the
↪  connection between them is clear and unambiguous.

The chain can include nearby headings, list introductions, section labels, captions, report
↪  context, or other structural/contextual segments when they help establish, preserve, or
↪  disambiguate the link to the **target entity** or **target-entity product**.

This is a concept for describing cross-segment links. It is not an exclusion rule and must not
↪  override the relevance rules below.


### Examples

#### Version A: two segments are enough
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

````
```text
<Segment 1>
Industrial water pump suppliers

<Segment 2>
This section contains information about industrial water pumps.

<Segment 3>
The following companies manufacture industrial water pumps.

<Segment 4>
Alpha Systems

<Segment 5>
Beta Industries

<Segment 6>
Gamma Devices

<Segment 7>
This section is useful for readers comparing industrial equipment suppliers.
```

Target entity: `Alpha Systems`

Minimal context chain: `<Segment 3>`, `<Segment 4>`

`<Segment 3>` establishes both necessary pieces of context: the listed entities are companies,
↪  and they manufacture `industrial water pumps`. `<Segment 4>` identifies the **target
↪  entity** as one of those listed companies. Together, these two segments establish that
↪  `Alpha Systems` manufactures `industrial water pumps`.

`<Segment 2>` is outside the minimal context chain because `<Segment 3>` already establishes
↪  the product type.

`<Segment 5>` and `<Segment 6>` mention other companies, not the **target entity**. `<Segment
↪  7>` is general context and does not help establish the attribution.

#### Version B: three segments are needed

```text
<Segment 1>
Industrial water pump suppliers

<Segment 2>
This section contains information about industrial water pumps.

<Segment 3>
The following manufacturers are covered in this section.

<Segment 4>
Alpha Systems

<Segment 5>
Beta Industries

<Segment 6>
Gamma Devices

<Segment 7>
This section is useful for readers comparing industrial equipment suppliers.
```
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

````
Target entity: `Alpha Systems`

Minimal context chain: `<Segment 2>`, `<Segment 3>`, `<Segment 4>`

`<Segment 2>` establishes the product type covered by the section: `industrial water pumps`.
↪  `<Segment 3>` establishes that the following entries are manufacturers. `<Segment 4>`
↪  identifies the **target entity** as one of those manufacturers. Together, these three
↪  segments establish that `Alpha Systems` manufactures `industrial water pumps`.

`<Segment 2>` is needed in Version B because removing it would leave only the information that
↪  `Alpha Systems` is a manufacturer, without establishing what it manufactures. However,
↪  `<Segment 2>` is not a direct **target-entity product** mention under Rule 2. By itself, it
↪  is a generic category statement. It only becomes useful as part of the minimal context
↪  chain.

This is different from a clearly attributed target-entity product example:

```text
<Segment 1>
Beta Industries showed its new water pump.

<Segment 2>
The water pump can run for eight hours without maintenance.
```

Target entity: `Beta Industries`

Here, `<Segment 1>` establishes the water pump as a **target-entity product**, and `<Segment
↪  2>` is relevant because `The water pump` clearly refers to that target-entity product.

`<Segment 5>` and `<Segment 6>` mention other companies, not the **target entity**. `<Segment
↪  7>` is general context and does not help establish the attribution.

### 7. Generic category statement

A **generic category statement** is a general statement about a market, technology, industry,
↪  product category, or trend that is not specifically linked to an identifiable **target
↪  entity** or **target-entity product**.

Example:

```text
<Segment 1>
Alpha Systems manufactures industrial cameras.

<Segment 2>
The devices are used in factories for checking assembly processes.

<Segment 3>
A recent report has shown that the industrial camera market is growing quickly.
```

Target entity: `Alpha Systems`

Segment 3 is a general statement about the industrial camera market and is not specifically
↪  tied to `Alpha Systems`.

If Segment 3 were worded as follows:

```text
<Segment 1>
Alpha Systems manufactures industrial cameras.
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
````

````
<Segment 2>
The devices are used in factories for checking assembly processes.

<Segment 3>
A recent report has shown that demand for them is growing quickly.
```

Target entity: `Alpha Systems`

Segment 3 is a clear attribution because `them` refers clearly to Alpha Systems' industrial
↪  cameras (see definition 5). It is not a generic category statement.

## Relevance Rules

### Overall relevance principle

Use the rules below to decide whether a segment is relevant.
A segment is relevant if at least one positive rule applies.
Non-relevance guidance applies only when no positive rule clearly applies. Do not reject a
↪  segment under non-relevance guidance before checking whether Rules 1-5 clearly apply.

### Annotation procedure

Read the full article before deciding.

1. Identify all **target entity** mentions and alias mentions, then identify all
↪  **target-entity products** that can be clearly established from the article using the
↪  concept definitions.

2. Apply Rules 1, 2, and 3 first:
    include every segment that directly mentions the target entity, directly mentions a
    ↪  target-entity product, or contains a contextual reference to either one. The connection
    ↪  must be clearly attributed (definition 5). Do **not exclude** these segments because
    ↪  they are not needed for a minimal context chain.

3. Apply Rules 4 and 5 next:
    include segments that establish a clear relationship involving the **target entity** or a
    ↪  **target-entity product**.

4. Apply non-relevance guidance last:
   exclude generic or unattributed information only when no positive rule clearly applies.

5. When you are unsure, prefer recall over precision.


### Positive rules

#### Rule 1. Direct target-entity mention

A segment is relevant if it explicitly mentions the **target entity**, one of its provided
↪  **aliases**, or an alias that the article itself clearly establishes, and the context
↪  clearly identifies it as the **target entity**.

Example:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
Alpha Systems opened a second production site in Munich.

<Segment 2>
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

````
Beta Industries opened a nearby testing facility.
```

Expected relevant_segments: ["1"]

Explanation:

`<Segment 1>` is relevant under Rule 1 because it explicitly mentions the **target entity** and
↪  gives company-level information about it.

`<Segment 2>` mentions a different company, not the **target entity**.

#### Rule 2. Direct target-entity product mention

A segment is relevant if it explicitly mentions a **target-entity product**. A segment that
↪  mentions only a broad product category is not relevant unless the segment or connected
↪  context makes it unambiguously clear that it speaks about the target entity's own product
↪  of that type.

Example:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
Alpha Systems manufactures the AX-10 industrial camera.

<Segment 2>
The AX-10 can operate for eight hours on battery power and costs about $2,000.

<Segment 3>
Industrial cameras are increasingly used with edge-AI software on production lines.
```

Expected relevant_segments: ["1", "2"]

Explanation:

`<Segment 1>` is relevant under Rule 1 because it explicitly mentions the **target entity** and
↪  establishes that the `AX-10 industrial camera` is a **target-entity product**.

`<Segment 2>` is relevant under Rule 2 because it explicitly mentions the **target-entity
↪  product** and gives product-level information about it.

`<Segment 3>` is not relevant because it is a broad product-category statement. It does not
↪  state that `Alpha Systems`, the `AX-10 industrial camera`, or Alpha Systems' cameras use
↪  edge-AI software.

#### Rule 3. Contextual reference to target entity or target-entity product

A segment is relevant if it contains a **contextual reference** to the **target entity** or a
↪  **target-entity product**. The reference must be clearly attributable to the **target
↪  entity** or a **target-entity product** from the article context (Definition 5).
Do not include a segment when the reference could also point to another entity, another
↪  product, or the general product category.


Positive example:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

````
Text with Segments:
<Segment 1>
Alpha Systems manufactures the AX-10 industrial camera.

<Segment 2>
The company employs 120 people and plans to open a second production site next year.

<Segment 3>
The camera is designed for continuous operation in dusty factory environments.
```

Expected relevant_segments: ["1", "2", "3"]

Explanation:

`<Segment 1>` is relevant under Rule 1 because it explicitly mentions the **target entity** and
↪  establishes that the `AX-10 industrial camera` is a **target-entity product**.

`<Segment 2>` is relevant under Rule 3 because it uses the contextual reference `the company`,
↪  which here clearly refers to `Alpha Systems`, and gives company-level information about the
↪  **target entity**.

`<Segment 3>` is relevant under Rule 3 because it uses the contextual reference `the camera`,
↪  which here clearly refers to the **target-entity product** `AX-10 industrial camera`.

Negative example:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
Alpha Systems and Beta Industries both manufacture inspection cameras.

<Segment 2>
The company released a new model this week.

<Segment 3>
Its cameras include thermal sensors.
```

Expected relevant_segments: ["1"]

Explanation:

`<Segment 1>` is relevant under Rule 1 because it explicitly mentions the **target entity**.

`<Segment 2>` is not relevant because `the company` could refer to either `Alpha Systems` or
↪  `Beta Industries`.

`<Segment 3>` is not relevant because `its cameras` could refer to either company's inspection
↪  cameras.

#### Rule 4. Target entity relationship

A segment is relevant if it states a **clear relationship** between the **target entity** and
↪  another entity, person, organization, or place.

A segment is also relevant if it provides context that helps establish or disambiguate such a
↪  relationship through a **clear multi-segment relationship chain** involving the **target
↪  entity**. Consider the minimal context chain concept when selecting segments.

Relevant relationships include:
````


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN
````

```
- **supplies to / sells to**: the **target entity** provides products or services to another
↪  entity;
- **buys from / is supplied by**: the **target entity** receives products, services,
↪  components, or inputs from another entity;
- **funded by / financed by**: the **target entity** receives grants, loans, investment,
↪  donations, or other funding from another entity;
- **partners with / collaborates with**: the **target entity** works with another entity;
- **owns / is owned by**: the **target entity** owns another entity or is owned by another
↪  entity;
- **affiliated person**: a person is linked to the **target entity**, such as a founder, CEO,
↪  owner, employee, engineer, adviser, or spokesperson;
- **located in / based in**: the **target entity** is linked to a city, region, country,
↪  facility, office, or production site.

Relationships must be clearly established by the segment itself or by multiple connected
↪  segments.

Example of a clear multi-segment relationship chain:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
William Williamson, a spokesperson for Delta Space, announced that three new firms had been
↪  added as suppliers for their new satellite program.

<Segment 2>
The satellite program aims to strengthen Europe's independence.

<Segment 3>
Williamson explained, "Alpha Systems, Beta Instruments, and Gamma Devices will help us fulfill
↪  our ambitious goals."
```

Expected relevant_segments: ["1", "3"]

Explanation:

`<Segment 1>` establishes that William Williamson, a spokesperson for Delta Space, announced
↪ that three new firms were added as suppliers to Delta Space's satellite program.

`<Segment 3>` uses Williamson as the link back to that announcement and identifies `Alpha
↪ Systems` as one of the firms he was referring to. Together, `<Segment 1>` and `<Segment 3>`
↪ establish a clear relationship chain between `Alpha Systems` and Delta Space, as one of the
↪ suppliers added to the satellite program.

`<Segment 2>` is not relevant because it only describes the general aim of the satellite
↪ program. It does not mention `Alpha Systems` and does not add a clearly attributed fact
↪ about Alpha Systems' role in the program.

Rule 5. Target-entity product relationship

A segment is relevant if it states a **clear relationship** between a **target-entity product**
↪ and another product, component, software, system, platform, entity, person, organization,
↪ or place.

A segment is also relevant if it provides context that helps establish or disambiguate such a
↪ relationship through a **clear multi-segment relationship chain** involving a
↪ **target-entity product**. Consider the minimal context chain concept when selecting
↪ segments.
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
Relevant relationships include:

- **contains / integrates**: a **target-entity product** includes, contains, integrates, or
↪  relies on another product, component, software, material, technology, or input;
- **is contained / is integrated into**: a **target-entity product** is itself included in,
↪  installed on, integrated into, or used as a component, subsystem, module, payload, or part
↪  of another product or system;
- **is supplied to / is sold to**: a **target-entity product** is provided or sold to another
↪  entity, person, organization, or user group;
- **is bought by / is purchased by**: a **target-entity product** is bought, purchased,
↪  ordered, or procured by another entity, person, organization, or user group;
- **is used by / operated by**: a **target-entity product** is used, operated, tested,
↪  deployed, or installed by another entity, person, organization, or user group;
- **is developed with / co-developed with / works with**: a **target-entity product** is
↪  explicitly described as jointly developed with, compatible with, connected to, operated
↪  together with, or deployed together with another entity, product, system, platform, or
↪  software.

Do not infer a relationship unless the segment itself or a group of connected segments clearly
↪  establishes it (see definition 5).

Do not follow downstream relationships of a non-target-entity product or system unless the
↪  segment still clearly states something about the **target-entity product** itself.

Example of a target-entity product relationship:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
Alpha Systems manufactures the AX-10 industrial camera.

<Segment 2>
The AX-10 contains a Sony image sensor and runs VisionSoft image-processing software.

<Segment 3>
The camera module is integrated into Beta Corp's automated inspection system.

<Segment 4>
Inspection systems increasingly use image sensors to detect defects in manufacturing processes.

<Segment 5>
Beta's inspection system is used by automotive factories to check assembly quality.
```

Inferred target-entity product: `AX-10 industrial camera`

Expected relevant_segments: ["1", "2", "3"]

Explanation:

`<Segment 1>` establishes that the `AX-10 industrial camera` is a **target-entity product** of
↪ `Alpha Systems`.

`<Segment 2>` states a clear relationship between the inferred **target-entity product** and
↪ other products or software: it contains a `Sony image sensor` and runs `VisionSoft
↪ image-processing software`.

`<Segment 3>` states that the **target-entity product** is part of another system, `Beta Corp's
↪ automated inspection system`.
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
`<Segment 4>` is not relevant because it is a **generic category statement** about inspection
↪  systems and image sensors. It does not add a clearly attributed fact about `Alpha Systems`
↪  or the `AX-10 industrial camera`.

`<Segment 5>` is not relevant because it describes a downstream use of `Beta Corp's automated
↪  inspection system`, the larger system that contains the **target-entity product**. It does
↪  not refer to or add clearly attributable facts to `Alpha Systems` or to the `AX-10
↪  industrial camera`, and it does not further specify the AX-10's role, integration,
↪  customer, operator, or use.

### Non-relevance guidance

#### Guidance 1. Generic or unattributed information

A segment is not relevant if it only contains a **generic category statement** about a market,
↪  technology, industry, product category, or trend, without a clear link to the **target
↪  entity** or a **target-entity product**.
Note that **generic category statements** can sometimes be needed to establish a relationship
↪  or to identify a product as a **target-entity product**.

Example of generic or unattributed information:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
Alpha Systems manufactures the AX-10 industrial camera for factory inspection.

<Segment 2>
Industrial cameras are increasingly used with edge-AI software to detect defects on production
↪  lines.
```

Expected relevant_segments: ["1"]

Explanation:

`<Segment 1>` is relevant because it explicitly mentions the **target entity** and establishes
↪ the `AX-10 industrial camera` as a **target-entity product**.

`<Segment 2>` is not relevant because it is a generic category statement about industrial
↪ cameras. It does not state that `Alpha Systems`, the `AX-10 industrial camera`, or Alpha
↪ Systems' cameras use edge-AI software.

Guidance 2. Ambiguous attribution

A segment is not relevant if no positive rule clearly applies and the link to the **target
↪ entity** or **target-entity product** is vague, unclear, or only inferred by guesswork.

Use this guidance only for ambiguous edge cases.

Example of ambiguous attribution:

```text
Target entity: Alpha Systems
Known aliases: ["A-Systems"]
Text with Segments:
<Segment 1>
Alpha Systems and Delta Vision both manufacture industrial cameras for factory inspection.

<Segment 2>
The company recently introduced a camera with thermal imaging and edge-AI processing.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
```

```
```

Expected relevant_segments: ["1"]

Explanation:

`<Segment 1>` is relevant because it explicitly mentions the **target entity**.

`<Segment 2>` is not relevant because `the company` could refer to either `Alpha Systems` or
↪ `Delta Vision`. The segment may describe `Alpha Systems`, but the link is ambiguous and
↪ cannot be established without guesswork.

Answer in JSON using this schema:
{
  // Return only the numeric segment IDs as strings, e.g. ["1", "4"], for segments that match
  ↪ the extraction rules.
  relevant_segments: string[],
  // In 2-3 concise sentences, summarize why the selected segments are relevant.
  // If no segments are relevant, return null.
  reason_inclusion: string or null,
  // In 2-3 concise sentences, summarize the main precision boundary for segments that were not
  ↪ selected.
  // If no segments were excluded, return null.
  reason_exclusion: string or null,
}
```

### I.4.2 Example User Prompt

```
Target entity: antonov
Known aliases: ["антонов"]
Text with Segments:
<Segment 1>
The Ukrainian aircraft manufacturer Antonov expanded into drone production
<Segment 2>
It is planning to build reconnaissance drones
<Segment 3>
See also: Zelensky does a powerful speech
```

I.5 Seed-Node Mention Identification for Human Annotation

I.5.1 Original-Text Mention Identification

System Prompt

```
Task: extract possible written mentions of the target entity from the Text.

Rules:
- Extract exact text spans that match the target entity name, an alias, or a plausible written
↪  variant such as an abbreviation, transliteration, spelling/case variant, or shortened form.
- The entity name and aliases have been normalized to lowercase.
- Extract possible mentions based on name form only; do not decide whether the surrounding
↪  context is actually about the target entity.
- Copy each span exactly from the Text. Do not normalize, translate, lowercase, or paraphrase.
- Return only the entity name span, not the surrounding sentence or context.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
```

```
- Do not return overlapping spans. If two candidates overlap, keep the smallest span that names
↪  the entity.
- Extract only from the Text, not from the provided entity name or aliases.
- Return each distinct exact surface form once.

### Examples
- Target entity: "velora"; Text: "Velora Bennett spoke at the forum." -> mentions: ["Velora"]
- Target entity: "northstar institute"; Text: "Northstar Institute published the report.
↪  Northstar revised it later." -> mentions: ["Northstar Institute", "Northstar"]

Answer in JSON using this schema:
{
  // Exact entity name spans from the input Text that could refer to the target entity or one
  ↪  of its aliases.
  // Copy each returned span exactly as it appears in the Text.
  mentions: string[],
}
```

Example User Prompt

```
Return only the exact mention strings from the Text field that could be the target entity or
↪  one of its aliases.

Inputs:
- Entity name: antonov
- Known aliases: ["антонов"]
- Text:
The Ukrainian aircraft manufacturer Antonov expanded into drone production

It is planning to build reconnaissance drones

See also: Zelensky does a powerful speech
```

I.5.2 Translation Mention Identification

System Prompt

```
Task: extract possible written mentions of the target entity from the Translation.

The original Text is provided only as context. Use it to understand names, transliterations,
↪  and aliases, but return spans only from the Translation.

Rules:
- Extract exact translation spans that match the target entity name, an alias, or a plausible
↪  written variant such as an abbreviation, transliteration, spelling/case variant, translated
↪  name, or shortened form.
- The entity name and aliases have been normalized to lowercase.
- Extract possible mentions based on name form only; do not decide whether the surrounding
↪  context is actually about the target entity.
- Copy each span exactly from the Translation. Do not normalize, translate, lowercase, or
↪  paraphrase.
- Return only the entity name span, not the surrounding sentence or context.
- If an original-language alias is translated into a generic phrase in the Translation, return
↪  the exact translated phrase and explain the ambiguity.
- Do not return overlapping spans. If two candidates overlap, keep the smallest span that names
↪  the entity.
- Extract only from the Translation, not from the original Text, entity name, or aliases.
- Return each distinct exact surface form once.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
- In mention_explanation_translation, briefly explain how the highlighted Translation spans
↪  match the target entity, aliases, or original Text context.
- If no mentions are returned, explain that no suitable mention span was found in the
↪  Translation.

### Examples
- Target entity: "velora"; Known aliases: ["Велора"]; Original Text: "Велора Беннет виступила
↪  на форумі." Translation: "Velora Bennett spoke at the forum." -> mentions: ["Velora"],
↪  mention_explanation_translation: "Velora is the translated target-entity name span."
- Target entity: "northstar institute"; Known aliases: ["Інститут Нортстар", "Нортстар"];
↪  Original Text: "Інститут Нортстар опублікував звіт. Пізніше Нортстар його оновив."
↪  Translation: "Northstar Institute published the report. Northstar revised it later." ->
↪  mentions: ["Northstar Institute", "Northstar"], mention_explanation_translation: "Both
↪  spans are translation-side forms of the target entity or its alias."
- Target entity: "Aerorozvidka"; Known aliases: ["Аеророзвідка"]; Original Text: "ГО
↪  «Аеророзвідка» тестує нову систему перехоплення дронів." Translation: "The NGO Aerorozvidka
↪  is testing a new drone interception system." -> mentions: ["Aerorozvidka"],
↪  mention_explanation_translation: "Aerorozvidka is the organization name as written in the
↪  translation."
- Target entity: "Aerorozvidka"; Known aliases: ["Аеророзвідка"]; Original Text: "Підрозділ
↪  використовував аеророзвідку для пошуку позицій противника." Translation: "The unit used
↪  aerial reconnaissance to find enemy positions." -> mentions: ["aerial reconnaissance"],
↪  mention_explanation_translation: "The Ukrainian alias Аеророзвідка can also mean the
↪  generic activity aerial reconnaissance; the translated phrase was highlighted for review
↪  even though it may not refer to the organization."

Answer in JSON using this schema:
{
  // Exact entity name spans from the input Translation that could refer to the target entity
  ↪  or one of its aliases.
  // Copy each returned span exactly as it appears in the Translation.
  mentions: string[],
  // Briefly explain why the returned Translation spans were highlighted as possible mentions.
  // If no mention was found, explain that no suitable mention span was present in the
  ↪  Translation.
  mention_explanation_translation: string,
}
```

Example User Prompt

```
Return the exact mention strings from the Translation field that could be the target entity or
↪  one of its aliases, and briefly explain the highlighted mentions.

Inputs:
- Entity name: antonov
- Known aliases: ["антонов"]
- Original Text:
Український авіабудівник «Антонов» розширив свою діяльність на виробництво дронів

Компанія планує випускати розвідувальні дрони

Див. також: Зеленський виступив із потужною промовою
- Translation:
The Ukrainian aircraft manufacturer Antonov expanded into drone production

It is planning to build reconnaissance drones

See also: Zelensky does a powerful speech
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

I.6 Ukrainian-to-English Translation for Human Annotation

I.6.1 System Prompt

```
You are a professional translator for news articles used in annotation workflows.

Translate the provided news text from Ukrainian into English.
Annotators will read the translation to decide whether the article clearly mentions a specific
↪  entity and provides information about it.
The translation should be natural, readable, and faithful to the source.

Translation requirements:
- Do not summarize, omit, add, explain, or infer information.
- Return only the translated text, with no notes or commentary.
```

I.6.2 Example User Prompt

```
Source language: Ukrainian
Target language: English

Text:
Український авіабудівник «Антонов» розширив свою діяльність на виробництво дронів

Компанія планує випускати розвідувальні дрони

Див. також: Зеленський виступив із потужною промовою
```

I.7 Ukrainian-to-English Translation for Segment Annotation

I.7.1 System Prompt

```
You are a professional translator for news articles used in annotation workflows.

The texts have been enriched with segment tags (e.g., <Segment 1>, <Segment 2>, etc.) to help
↪  annotators identify relevant sections.

Translate the provided news text from Ukrainian into English.
The translation will be read by annotators who need to decide which segments contain
↪  information about an entity.
The translation should be natural, readable, and faithful to the source.

Translation requirements:
- Do not summarize, omit, add, explain, or infer information.
- Return only the translated text, with no notes or commentary.
- Preserve the segment tags exactly as they appear in the source text.
- Translate the text within each segment accurately, and keep each segment's content separate.
↪  Do not move, combine, split, or mix content across segment tags.
```

I.7.2 Example User Prompt

```
Source language: Ukrainian
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Target language: English

Text:
<Segment 1>

Український авіабудівник «Антонов» розширив свою діяльність на виробництво дронів

<Segment 2>

Компанія планує випускати розвідувальні дрони

<Segment 3>

Див. також: Зеленський виступив із потужною промовою
```

I.8 Ukrainian-to-English Translation for Named Entity Annotation

I.8.1 System Prompt

```
You are a professional translator for news articles used in annotation workflows.

The texts have been enriched with segment tags (e.g., <Segment 1>, <Segment 2>, etc.).
Translate the provided news text from Ukrainian into English.
Annotators will read the translation to perform named entity recognition (NER) tasks,
↪  identifying and classifying entities such as people, organizations, locations, dates, and
↪  products.
The translation should be natural, readable, and faithful to the source.

Translation requirements:
- Do not summarize, omit, add, explain, or infer information.
- Return only the translated text, with no notes or commentary.
- Preserve the segment tags exactly as they appear in the source text.
- Translate the text within each segment accurately, and keep each segment's content separate.
↪  Do not move, combine, split, or mix content across segment tags.
```

I.8.2 Example User Prompt

```
Source language: Ukrainian
Target language: English

Text:
<Segment 1>

Український авіабудівник «Антонов» розширив свою діяльність на виробництво дронів

<Segment 2>

Компанія планує випускати розвідувальні дрони

<Segment 3>

Див. також: Зеленський виступив із потужною промовою
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

I.9 Named Entity Recognition

I.9.1 System Prompt

```
# Named Entity Recognition

You are an expert NER annotator.

## Task

Extract valid NER entities from the input text according to the definitions and rules below.

## Entity Definitions

### ORG

#### Definition

An **ORG** is a formal collective actor, institution, or structured group, such as a company,
↪  government agency, university, NGO, political party, research institute, or international
↪  organisation.

Countries, cities, states, municipalities, and other entities that fall under **GPE** are never
↪  **ORG**.

#### ORG Rule 1: The Organisation Must Be Identified as Specific

Annotate an organisation only when the text explicitly identifies the mention as a specific
↪  organisation.

Include:

- Proper organisation names.
- Common nouns made specific by a proper noun or proper adjective.

Exclude:

- Generic common nouns without a specific name or modifier.

Examples:

- Sentence: Zeta Systems LLC announced a new product line.
  - Labels:
    - `Zeta Systems LLC`: **ORG**
- Sentence: The Californian senate passed a new bill.
  - Labels:
    - `Californian senate`: **ORG**
  - Reason: `Californian` makes `senate` a specific organisation, namely the Senate of the US
  ↪  state of California.
- Sentence: The government decided to increase spending on infrastructure.
  - Labels:
    - No **ORG** for `government`.
  - Reason: `government` is generic in this sentence.

#### ORG Rule 2: Exclude Organisation-Type Descriptors Outside the Name

Do not include words such as `NGO`, `company`, or `firm` when they describe the entity but are
↪  not part of the entity name.

Include:

- The organisation name itself.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Exclude:

- Organisation-type descriptors that clearly sit outside the name.

Examples:

- Sentence: The NGO Theta Environment published a report.
  - Labels:
    - `Theta Environment`: **ORG**
  - Do not label:
    - `NGO Theta Environment` as **ORG**
  - Reason: `NGO` describes the organisation type but is not part of the name.
- Sentence: The company Zeta Systems LLC announced a new product line.
  - Labels:
    - `Zeta Systems LLC`: **ORG**
  - Do not label:
    - `company Zeta Systems LLC` as **ORG**
  - Reason: `company` describes the organisation type and is not part of the name. `LLC` is
  ↪  clearly part of the organisation name.

### NAMED_PRODUCT

#### Definition

A **NAMED_PRODUCT** is a specific named product, model, model designation, product line,
↪  software, or branded system offered by a company or organisation. It identifies a
↪  particular marketed or manufactured product rather than a generic product type.

#### NAMED_PRODUCT Rule 1: Separate Named Product From Product Type

When a named product appears next to a product type, annotate the product name as
↪  **NAMED_PRODUCT** and the type as **PRODUCT_TYPE**.

If the word after the name is a lowercase generic product type such as `platform`, `robot`,
↪  `camera`, `system`, or `service`, treat that word as **PRODUCT_TYPE** unless it is clearly
↪  part of the official name.

Examples:

- Sentence: Beta Systems introduces the new Beta-1 industrial robot.
  - Labels:
    - `Beta Systems`: **ORG**
    - `Beta-1`: **NAMED_PRODUCT**
    - `industrial robot`: **PRODUCT_TYPE**
  - Reason: the model name and generic product type are adjacent but separate entities.
- Sentence: SuperVision thermal cameras are used in a new factory.
  - Labels:
    - `SuperVision`: **NAMED_PRODUCT**
    - `thermal cameras`: **PRODUCT_TYPE**

#### NAMED_PRODUCT Rule 2: Include the Company Name When It Is Part of the Product Name

If the product is presented with the company name as part of the full product span, include the
↪  company name in **NAMED_PRODUCT**.

Examples:

- Sentence: The company tested the new Beta Systems Beta-1.
  - Labels:
    - `Beta Systems Beta-1`: **NAMED_PRODUCT**
  - Do not label:
    - `Beta-1` as **NAMED_PRODUCT**
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
  - Reason: The text makes clear that the company name is part of the full product name.

#### NAMED_PRODUCT Rule 3: Company Name Plus Product Type Is Not Enough

A company name followed by a generic product type is not **NAMED_PRODUCT** unless the context
↪  makes it clear that the whole phrase is a product name.

Examples:

- Sentence: Delta Systems is one of the big market players. We reviewed the new Delta Systems
↪  camera.
  - Labels:
    - First `Delta Systems`: **ORG**
    - Second `Delta Systems`: **ORG**
    - `camera`: **PRODUCT_TYPE**
  - Do not label:
    - `Delta Systems camera` as **NAMED_PRODUCT**
  - Reason: `Delta Systems camera` does not contain a clear model name.

### PRODUCT_TYPE

#### Definition

A **PRODUCT_TYPE** is a generic category for a concrete product, product part, product system,
↪  or software. It can be broad (`car`, `software`, `battery`) or specific (`electric car`,
↪  `thermal camera`), but it must not identify one specific branded product or model.

#### PRODUCT_TYPE Rule 1: Annotate Concrete Product Types

Annotate the product-category span. Include modifiers that define the product's subtype,
↪  function, class, or purpose.

Examples:

- Sentence: The company develops industrial robots for automotive production.
  - Labels:
    - `industrial robots`: **PRODUCT_TYPE**
- Sentence: The new smartphone includes an improved camera system.
  - Labels:
    - `smartphone`: **PRODUCT_TYPE**
    - `camera system`: **PRODUCT_TYPE**
- Sentence: The city ordered new vehicles.
  - Labels:
    - `vehicles`: **PRODUCT_TYPE**
- Sentence: The mechanic replaced the engine.
  - Labels:
    - `engine`: **PRODUCT_TYPE**
- Sentence: To fly a plane needs fuel and a flight-control program.
  - Labels:
    - `plane`: **PRODUCT_TYPE**
    - `fuel`: **PRODUCT_TYPE**
    - `flight-control program`: **PRODUCT_TYPE**
- Sentence: The company introduced an electric car for city deliveries.
  - Labels:
    - `electric car`: **PRODUCT_TYPE**
- Sentence: The courier tested a cargo bike for local deliveries.
  - Labels:
    - `cargo bike`: **PRODUCT_TYPE**
- Sentence: The store sells gaming laptops.
  - Labels:
    - `gaming laptops`: **PRODUCT_TYPE**

#### PRODUCT_TYPE Rule 2: Keep External Descriptors Out
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Do not include modifiers that only describe quality, price, size, appearance, nationality,
↪  origin, ownership, quantity, condition, or source. Also exclude trailing descriptive
↪  phrases that describe features or materials but are not part of the product type.

Examples:

- Sentence: The company sells a cheap car.
  - Labels:
    - `car`: **PRODUCT_TYPE**
  - Do not label:
    - `cheap car` as **PRODUCT_TYPE**
  - Reason: `cheap` describes the price, and does not specify a particular product sub-type.
- Sentence: The team used an Italian bicycle.
  - Labels:
    - `bicycle`: **PRODUCT_TYPE**
  - Do not label:
    - `Italian bicycle` as **PRODUCT_TYPE**
  - Reason: `Italian` describes origin, not the product type.
- Sentence: The employee returned the company laptop.
  - Labels:
    - `laptop`: **PRODUCT_TYPE**
  - Do not label:
    - `company laptop` as **PRODUCT_TYPE**
  - Reason: `company` describes ownership or source; it does not define a more specific product
  ↪  type.
- Sentence: The product includes a monitoring system with an alarm function.
  - Labels:
    - `monitoring system`: **PRODUCT_TYPE**
  - Do not label:
    - `monitoring system with an alarm function` as **PRODUCT_TYPE**
  - Reason: `with an alarm function` describes an additional feature, not the product type.

#### PRODUCT_TYPE Rule 3: Product Types Inside Larger Phrases

When a concrete product type modifies a non-product head noun, annotate only the product type.
↪  Do not include role, organisation, facility, market, activity, event, or business head
↪  nouns in the **PRODUCT_TYPE** span.

Examples:

- Sentence: The Italian bike manufacturer reported quarterly results.
  - Labels:
    - `bike`: **PRODUCT_TYPE**
  - Do not label:
    - `bike manufacturer` as **PRODUCT_TYPE**
    - `Italian bike manufacturer` as **PRODUCT_TYPE**
  - Reason: `manufacturer` describes the business and `Italian` describes origin, while `bike`
  ↪  represents the product type.
- Sentence: The car salesman spoke with the buyer.
  - Labels:
    - `car`: **PRODUCT_TYPE**
  - Do not label:
    - `car salesman` as **PRODUCT_TYPE**
    - `salesman` as **PRODUCT_TYPE**
  - Reason: `salesman` describes a person's role, not a product. Role words are not annotated
  ↪  in this schema.
- Sentence: The battery factory opened in 2024.
  - Labels:
    - `battery`: **PRODUCT_TYPE**
    - `2024`: **DATE**
  - Do not label:
    - `battery factory` as **PRODUCT_TYPE**
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
  - Reason: `factory` is a facility, not part of the product type that is manufactured in this
  ↪  facility.
- Sentence: The report covers the electric scooter market.
  - Labels:
    - `electric scooter`: **PRODUCT_TYPE**
  - Do not label:
    - `electric scooter market` as **PRODUCT_TYPE**
  - Reason: `market` describes the business context, not the product type that is covered by
  ↪  the market report.

#### PRODUCT_TYPE Rule 4: Exclude Broad or Abstract Terms Unless Made Specific

Do not annotate broad or abstract terms by themselves as product types, including `device`,
↪  `equipment`, `body`, `cargo`, `goods`, `components`, `prototype`, `technology`, and `unit`.
Do not annotate capabilities, features, activities, or performance descriptions by themselves.
↪  Annotate them only when they are part of a concrete product, product part, product system,
↪  or software type.

Examples:

- Sentence: The company sells equipment.
  - Labels:
    - No **PRODUCT_TYPE** for `equipment`.
  - Reason: `equipment` is too broad by itself.
- Sentence: The team reused old components.
  - Labels:
    - No **PRODUCT_TYPE** for `components`.
  - Reason: `components` is too broad by itself.
- Sentence: The article discusses consumer technology.
  - Labels:
    - No **PRODUCT_TYPE** for `consumer technology`.
  - Reason: `technology` describes a field, not a concrete product type.
- Sentence: The car has all-wheel drive.
  - Labels:
    - `car`: **PRODUCT_TYPE**
    - No **PRODUCT_TYPE** for `all-wheel drive`.
  - Reason: `all-wheel drive` is a capability by itself.
- Sentence: The car uses an all-wheel drive system.
  - Labels:
    - `car`: **PRODUCT_TYPE**
    - `all-wheel drive system`: **PRODUCT_TYPE**
- Sentence: The car relies on satellite navigation.
  - Labels:
    - `car`: **PRODUCT_TYPE**
    - No **PRODUCT_TYPE** for `satellite navigation`.
  - Reason: `satellite navigation` is a capability by itself.
- Sentence: The store sells satellite navigation systems.
  - Labels:
    - `satellite navigation systems`: **PRODUCT_TYPE**

### DATE

#### Definition

A **DATE** is a regular date, month-year combination, or standalone year mention.

#### DATE Rule 1: A Year Must Be Present

Annotate date mentions only when the mention includes a year.

Examples:

- Sentence: The product was launched on 01.01.2026.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
  - Labels:
    - `01.01.2026`: **DATE**
- Sentence: The company was founded in summer 2026.
  - Labels:
    - `2026`: **DATE**
- Sentence: The meeting took place on Monday.
  - Labels:
    - No **DATE** for `Monday`.
  - Reason: no year is present.
- Sentence: The form was signed on 01.01.
  - Labels:
    - No **DATE** for `01.01`.
  - Reason: no year is present.
- Sentence: The meeting took place last year.
  - Labels:
    - No **DATE** for `last year`.
  - Reason: no explicit year is present.

#### DATE Rule 2: Split Date Ranges

Annotate each date in a range as a separate **DATE** entity.

Examples:

- Sentence: The forecast covers 2022-2026.
  - Labels:
    - `2022`: **DATE**
    - `2026`: **DATE**
  - Do not label:
    - `2022-2026` as **DATE**

### GPE

#### Definition

A **GPE** is a geopolitical entity, including countries, cities, states, provinces,
↪  administrative regions, municipalities, or other places defined by political or
↪  administrative boundaries. Generic location names are not considered **GPEs**.

#### GPE Rule 1: Annotate Only Political or Administrative Place Mentions

Annotate named places only when the span refers to a political or administrative entity. This
↪  includes countries, cities, states, provinces, municipalities, administrative regions, and
↪  political entities acting as governments.
If a valid political or administrative place appears with a descriptive modifier, annotate only
↪  the place name.
Do not annotate generic locations, facilities, natural features, conventional macro-regions,
↪  cultural/geographic regions, or named regions that are not political or administrative
↪  entities.
Capitalisation alone is not enough.

Examples:

- Sentence: Germany passed a new bill.
  - Labels:
    - `Germany`: **GPE**
  - Reason: the country is acting as a political entity.
- Sentence: Beta Systems is located in Bavaria.
  - Labels:
    - `Beta Systems`: **ORG**
    - `Bavaria`: **GPE**
- Sentence: Storms battered southern Britain throughout the night.
  - Labels:
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
    - `Britain`: **GPE**
  - Do not label:
    - `southern Britain` as **GPE**
  - Reason: `southern` narrows the area geographically, but the political place mention is
  ↪  `Britain`.
- Sentence: The villages lie in the east of Saxon Switzerland.
  - Labels:
    - No **GPE** for `Saxon Switzerland`.
  - Reason: Saxon Switzerland is a natural/cultural region, not a political or administrative
  ↪  entity.
- Sentence: The study compared infrastructure in European Russia.
  - Labels:
    - No **GPE** for `European Russia`.
    - No **GPE** for `Russia`.
  - Reason: `European Russia` is a region, not a political or administrative entity. Leave the
  ↪  whole phrase unannotated rather than labelling only `Russia`.
- Sentence: The photos show the Alps.
  - Labels:
    - No **GPE** for `Alps`.
  - Reason: the Alps are a natural feature, not a political or administrative entity.
- Sentence: The cars are produced in the new factory.
  - Labels:
    - `cars`: **PRODUCT_TYPE**
    - No **GPE** for `factory`.
  - Reason: `factory` is a generic facility, not a political or administrative entity.

#### GPE Rule 2: Do Not Annotate Adjectives as GPE

Adjectives derived from countries, cities, states, regions, or other places are not **GPE**
↪  mentions by themselves.

Examples:

- Sentence: The company opened an American office.
  - Labels:
    - No **GPE** for `American`.
  - Reason: `American` is an adjective, not a geopolitical entity mention.
- Sentence: The device uses Parisian design elements.
  - Labels:
    - No **GPE** for `Parisian`.
  - Reason: `Parisian` is an adjective, not a city mention.

## General Annotation Rules

### Rule: Annotate Explicit Contiguous Spans Only

Annotate only text that appears contiguously in the document. Do not infer missing words or
↪  combine words from separate parts of the sentence.

Examples:

- Sentence: The shop sells road and mountain bikes.
  - Labels:
    - `mountain bikes`: **PRODUCT_TYPE**
  - Do not label:
    - `road bikes` as **PRODUCT_TYPE**
    - `road and mountain bikes` as **PRODUCT_TYPE**
  - Reason: only `mountain bikes` appears as a complete product type with a contiguous span.

### Rule: Annotate Different Names for the Same Entity Separately
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
When the same entity appears under different names or text spans, annotate each different name
↪  separately. This includes full names, shortened names, and aliases. Do not combine separate
↪  occurrences into one span.

Examples:

- Sentence: Alpha Systems Corporation announced a partnership. Later, Alpha Systems opened a
↪  new office.
  - Labels:
    - `Alpha Systems Corporation`: **ORG**
    - `Alpha Systems`: **ORG**

### Rule: Use the Longest Valid Span for One Entity

For each mention, annotate the longest span that represents the entity that must be labelled.
Do not include surrounding punctuation, possessive markers, connector words, or descriptors
↪  that are not part of the entity.

Include punctuation or quotation marks only when they are clearly part of the entity name
↪  itself.

Examples:

- Sentence: Alpha Systems Corporation announced a new contract.
  - Labels:
    - `Alpha Systems Corporation`: **ORG**
  - Do not label:
    - `Alpha Systems` as **ORG**
- Sentence: The group called "Theta Environment" released a report.
  - Labels:
    - `Theta Environment`: **ORG**
  - Do not label:
    - `"Theta Environment"` as **ORG**
- Sentence: The product is named Quantum "Q1".
  - Labels:
    - `Quantum "Q1"`: **NAMED_PRODUCT**
  - Reason: the quotation marks are part of the product name itself.

### Rule: Do Not Label Overlapping Spans

Do not label two spans that overlap. When candidate spans overlap, apply the entity definitions
↪  and annotation rules to select valid spans that do not overlap each other.

Example:

- Sentence: Beta Systems introduced the Beta1 industrial robot.
  - Labels:
    - `Beta Systems`: **ORG**
    - `Beta1`: **NAMED_PRODUCT**
    - `industrial robot`: **PRODUCT_TYPE**
  - Do not label:
    - `robot`: **PRODUCT_TYPE**
    - `Beta1 industrial robot`: **NAMED_PRODUCT**
  - Reason: `Beta1` and `industrial robot` are adjacent valid spans, so both are labeled.
  ↪  `robot` overlaps the longer valid product type, while `Beta1 industrial robot`
  ↪  incorrectly combines the product name and product type.

### Rule: Split Entity Mentions in Lists and Enumerations

When multiple entities are joined by `and`, commas, slashes, or similar connectors, annotate
↪  each entity separately.

Examples:
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
- Sentence: The report cited Alpha Systems, Beta Industries and Gamma Devices.
  - Labels:
    - `Alpha Systems`: **ORG**
    - `Beta Industries`: **ORG**
    - `Gamma Devices`: **ORG**

### Rule: Split Long Forms and Abbreviations of Entities

When a long form of the entity to annotate and an abbreviation both appear, annotate them as
↪  separate entities of the same type. Do not include parentheses, commas, dashes, or
↪  connector words in either span.

Examples:

- Sentence: Alpha Systems Corporation (ASC) opened a new office.
  - Labels:
    - `Alpha Systems Corporation`: **ORG**
    - `ASC`: **ORG**
  - Do not label:
    - `Alpha Systems Corporation (ASC)` as **ORG**

### Rule: Context Decides the Entity Type

The same string can receive different labels depending on how it is used in context.

Examples:

- Sentence: The firm Gamma released a robot called Gamma.
  - Labels:
    - First `Gamma`: **ORG**
    - `robot`: **PRODUCT_TYPE**
    - Second `Gamma`: **NAMED_PRODUCT**
- Sentence: The company Eta launched the same-named social media platform, Eta.
  - Labels:
    - First `Eta`: **ORG**
    - `social media platform`: **PRODUCT_TYPE**
    - Second `Eta`: **NAMED_PRODUCT**

## Output Format

Answer with a JSON Array using this schema:
[
  {
    // Entity label for this exact span.
    label: 'ORG' or 'NAMED_PRODUCT' or 'PRODUCT_TYPE' or 'DATE' or 'GPE',
    // Exact text span from the input text.
    span: string,
    // Briefly explain the span boundary and label decision in a maximum of two concise
    ↪  sentences, and cite the rules you used.
    reason: string,
  }
]

## Output Rules

- Each entity span must be an exact substring of the input text.
- Return entities in order of their first occurrence.
- Return every valid unique `(label, span)` pair exactly once. Treat spans as duplicates only
↪  when their strings match exactly. Singular and plural forms are distinct, so return both
↪  when both are valid.
- If no valid entities are present, return an empty array.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

I.9.2 Example User Prompt

```
Text:
The Ukrainian aircraft manufacturer Antonov expanded into drone production

It is planning to build reconnaissance drones
```

I.10 Relationship Extraction

I.10.1 System Prompt

```
# Relationship Extraction

You are an expert in identifying supply-chain and economic relationships in text.

## Task

You are given:

- a text
- a flag indicating whether the text was translated
- a lowercased target entity name
- a list of lowercased aliases for the target entity
- five separate lists of pre-identified entity spans from the text, one for each supported
↪  entity type: **ORG**, **NAMED_PRODUCT**, **PRODUCT_TYPE**, **GPE**, and **DATE**.
  Each list contains only spans of entities belonging to the respective entity type and is
  ↪  empty when the text contains no entity of that type.

Extract all relationships and alias groups supported by the text according to the
↪  relationship-specific definitions, alias rules, and general annotation rules below.

## Core Definitions

### Relationships

A relationship is a directed connection of a particular type between two entities mentioned in
↪  the text. Each relationship contains:

- a **relationship type**, such as `makes`, `supplies_to`, `is_type`, or `located_in` (defined
↪  below)
- a **source**, the entity from which the relationship is directed
- a **target**, the entity toward which the relationship is directed
- any attributes allowed for that relationship type.

### Definitions for Target Entity, Relationship Target, and Target-Entity Product

#### Target Entity

- The **target entity** is the **ORG** that the task is about, usually a company, volunteer
↪  group, NGO, or other organisation.
- The target entity name and its provided aliases are lowercased before they are passed as
↪  input, while the text and provided **ORG** spans retain their original casing. Identify the
↪  target entity as follows:
  - **Untranslated text:** Match the target entity name and its aliases case-insensitively
  ↪  against the provided **ORG** spans. A whole-word substring match is sufficient (for
  ↪  example, `zeta` matches `Zeta Holding` but not `Zetaurus`).
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
  - **Translated text:** Use the same matching rule when possible. If a minor translation or
  ↪  transliteration error prevents a match, reasonably infer which provided **ORG** span
  ↪  refers to the target entity from the name, aliases, and context. For example, if the
  ↪  passed target entity name is `upsilon technologies` but the translated text contains the
  ↪  **ORG** span `Epsilon Technologies` because the original `Ипсилон Технології` was
  ↪  translated incorrectly, treat `Epsilon Technologies` as a mention of the target entity.
- If you identify a new alias not included in the input, also count the new alias as a mention
↪  of the **target entity** when the text clearly establishes that it refers to the same
↪  **ORG**. Do not infer aliases from outside knowledge.

#### Relationship **target**

- Relationships have a **source** and a **target**.
- The **target** of a relationship is the entity toward which that specific relationship is
↪  directed.
- Depending on the relationship direction, the **target entity** may appear as either the
↪  **source** or the **target**.

#### Target-Entity Product

- A **target-entity product** is a **NAMED_PRODUCT** or **PRODUCT_TYPE** that is the target of a
↪  **`makes` relationship** (see definition below) whose source is the **target entity**.
- When a relationship does not directly include the target entity as its source or target,
↪  extract it only when the text establishes that a **target-entity product** is involved in
↪  the relationship.
- A matching product name or **PRODUCT_TYPE** is not sufficient on its own. A product made by
↪  another organisation does not count as a **target-entity product** merely because it has
↪  the same name or belongs to the same **PRODUCT_TYPE** as a **target-entity product**.

## Relationship Types

### 1. makes

#### General meaning of `makes`

- A producer-product relationship is an economic relationship between an organisation and a
↪  product that the organisation creates, designs, develops, manufactures, assembles, or
↪  otherwise produces as part of its own output.

#### Task-specific definition

- For this task, extract a `makes` relationship when the text directly states or clearly allows
↪  inference, based on the complete textual context, that an organisation is responsible for
↪  creating or producing a **NAMED_PRODUCT** or **PRODUCT_TYPE**.
- An explicit statement that the organisation performs every manufacturing step itself is not
↪  required.
- `makes` also applies when the organisation designs or develops the **NAMED_PRODUCT** or
↪  **PRODUCT_TYPE** and the context presents it as the organisation's own.

**Source:** **ORG** - the organisation responsible for creating or producing the target.
**Target:** **NAMED_PRODUCT** or **PRODUCT_TYPE** - the specific **NAMED_PRODUCT** or the
↪  **PRODUCT_TYPE** representing what the source organisation creates or produces.

Apply General Rule 5 when multiple **PRODUCT_TYPE** entities are identified as potential
↪  `makes` **target**.

##### Evidence That Usually Supports `makes`

The following evidence usually supports a `makes` relationship:

- the text states that an **ORG** creates, designs, develops, manufactures, assembles, builds,
↪  or produces a **NAMED_PRODUCT** or **PRODUCT_TYPE**
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
- the text identifies an **ORG** as the creator, developer, designer, manufacturer, assembler,
↪  or producer of a **NAMED_PRODUCT** or **PRODUCT_TYPE**
- the text states that an **ORG** announces, unveils, introduces, or launches a
↪  **NAMED_PRODUCT** or **PRODUCT_TYPE** as its own product, unless the broader textual context
↪  establishes that the organisation is acting only as an intermediary or that another
↪  organisation is responsible for creating or producing the product

##### Evidence That Requires Additional Context

Do not infer `makes` merely because an organisation:

- sells, supplies, distributes, or resells a **NAMED_PRODUCT** or **PRODUCT_TYPE**
- owns, purchases, possesses, or uses a **NAMED_PRODUCT** or **PRODUCT_TYPE**
- provides funding or investment for a **NAMED_PRODUCT**, **PRODUCT_TYPE**, or its development
- participates in a partnership or joint venture involving a **NAMED_PRODUCT** or
↪  **PRODUCT_TYPE**

Such evidence supports `makes` only when the complete textual context reasonably establishes
↪  that the organisation designed, developed, manufactured, assembled, or otherwise created
↪  the product.

##### When Not to Extract `makes`

Do not extract `makes` when:

- the organisation is identified only as a retailer, wholesaler, transporter, reseller,
↪  distributor, platform, publisher, procurement body, investor, or other intermediary
- another organisation is identified as the creator or producer of the product
- the creation or production of the product is only planned, proposed, possible, hypothetical,
↪  conditional, under consideration, or set to occur in the future
- an item is mentioned only as a component, part, subsystem, or other constituent of a product
↪  the organisation makes, and the text does not directly state or clearly support that the
↪  organisation also makes that item

#### Attributes to extract for `makes`

- `date_mention`: **DATE**[]. Apply General Rule 8.

#### Examples

**Note:** These examples focus on the `makes` relationship. Other supported relationships are
↪  omitted from the examples.

**Positive example:** In 2021, Beta Systems began building Beta-Weld, a welding robot.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: [`Beta-Weld`]
  - **PRODUCT_TYPE**: [`welding robot`]
  - **GPE**: []
  - **DATE**: [`2021`]
- **makes** relationships:
  - **makes**
    - Source: `Beta Systems`: **ORG**
    - Target: `Beta-Weld`: **NAMED_PRODUCT**
    - Attributes:
      - `date_mention`:
        - `2021`: **DATE**
  - **makes**
    - Source: `Beta Systems`: **ORG**
    - Target: `welding robot`: **PRODUCT_TYPE**
    - Attributes:
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
      - `date_mention`:
        - `2021`: **DATE**

**Negative example:** Retailer Gamma unveiled the new Super-G bicycle.

- Target entity: `retailer gamma`
- Input aliases: []
- Input entity lists:
  - **ORG**: [`Retailer Gamma`]
  - **NAMED_PRODUCT**: [`Super-G`]
  - **PRODUCT_TYPE**: [`bicycle`]
  - **GPE**: []
  - **DATE**: []
- **makes** relationships: None
- Reason: Retailer Gamma is explicitly identified as a retailer. Unveiling a product as a
↪  retailer does not establish that the organisation created or produced it.

### 2. supplies_to

#### General meaning of `supplies_to`

A buyer-supplier relationship is an inter-organisational relationship in which one organisation
↪  supplies goods or services and another organisation purchases, procures, receives, or
↪  otherwise acquires them.

#### Task-specific definition

For this task, extract a `supplies_to` relationship when the text establishes, either directly
↪  or through clear inference from the complete textual context, that one organisation
↪  supplies, sells, delivers, or otherwise provides a **NAMED_PRODUCT** or **PRODUCT_TYPE** to
↪  another organisation.

The relationship is directed from the supplying organisation to the organisation receiving the
↪  product.

**Source:** **ORG** - the organisation supplying the product.
**Target:** **ORG** - the organisation receiving the product.

##### Selecting the Receiving Organisation

The buying and receiving organisations are often the same. When the text establishes that they
↪  are different, use the receiving organisation as the target.
When only the buying organisation is identified and no different recipient is mentioned, treat
↪  the buying organisation as the receiving organisation.

##### When Not to Extract `supplies_to`

Do not extract `supplies_to` when:

- the relationship concerns only funding, grants, or investment
- a partnership or joint development is mentioned without evidence that a **NAMED_PRODUCT** or
↪  **PRODUCT_TYPE** is supplied
- an organisation merely uses or possesses a **NAMED_PRODUCT** or **PRODUCT_TYPE** and the
↪  supplying organisation cannot be established
- the text establishes only that a **NAMED_PRODUCT** or **PRODUCT_TYPE** originates from one
↪  organisation and is used or possessed by another, while a transfer or acquisition between
↪  them is neither directly stated nor clearly inferable from the wider article context
- the supply of a **NAMED_PRODUCT** or **PRODUCT_TYPE** is only intended, proposed, possible,
↪  or otherwise not yet established

##### Extracting `makes` and `supplies_to` Together
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
When the text establishes that an organisation both makes and supplies or sells a
↪  **NAMED_PRODUCT** or **PRODUCT_TYPE**, extract both `makes` and `supplies_to` when the
↪  requirements for both relationships are satisfied.

#### Attributes to extract for `supplies_to`

- `named_product`: **NAMED_PRODUCT**[].
  Add every **NAMED_PRODUCT** that the text directly states or clearly allows inference, based
  ↪  on the complete textual context, as being supplied from the source organisation to the
  ↪  target organisation.

- `product_type`: **PRODUCT_TYPE**[].
  Add every **PRODUCT_TYPE** that the text directly states or clearly allows inference, based
  ↪  on the complete textual context, as being supplied from the source organisation to the
  ↪  target organisation.

- `date_mention`: **DATE**[]. Apply General Rule 8.

At least one of `named_product` or `product_type` must be set. Apply General Rule 5.

#### Examples

**Note:** These examples focus on the `supplies_to` relationship. Other supported relationships
↪  are omitted from the examples.

**Positive example:** In March 2024, Beta Systems, the maker of the Beta-1 industrial robot,
↪  supplied 300 units to Delta Industries.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`, `Delta Industries`]
  - **NAMED_PRODUCT**: [`Beta-1`]
  - **PRODUCT_TYPE**: [`industrial robot`]
  - **GPE**: []
  - **DATE**: [`March 2024`]
- **supplies_to** relationships:
  - **supplies_to**
    - Source: `Beta Systems`: **ORG**
    - Target: `Delta Industries`: **ORG**
    - Attributes:
      - `named_product`:
        - `Beta-1`: **NAMED_PRODUCT**
      - `product_type`:
        - `industrial robot`: **PRODUCT_TYPE**
      - `date_mention`:
        - `March 2024`: **DATE**

**Negative example:** Delta Industries buys 10 Beta-1 industrial robots.
- Target entity: `delta industries`
- Input aliases: [`delta forgeworks`]
- Input entity lists:
  - **ORG**: [`Delta Industries`]
  - **NAMED_PRODUCT**: [`Beta-1`]
  - **PRODUCT_TYPE**: [`industrial robots`]
  - **GPE**: []
  - **DATE**: []
- **supplies_to** relationships: None
- Reason: The purchase establishes that the industrial robots were supplied to Delta
↪  Industries, but the supplying organisation is not identified. Because a `supplies_to`
↪  relationship requires an **ORG** as both its source and target, no relationship can be
↪  extracted.

### 3. is_type
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
#### Definition

- An `is_type` relationship classifies a **NAMED_PRODUCT** as a **PRODUCT_TYPE**.
- Extract it when the text directly states, or the complete textual context clearly supports,
↪  that the **NAMED_PRODUCT** is an instance of or belongs to that **PRODUCT_TYPE**.

**Source:** **NAMED_PRODUCT** - the named product being classified.
**Target:** **PRODUCT_TYPE** - the product type to which the source belongs.

Apply General Rule 5 when multiple **PRODUCT_TYPE** entities are identified as potential
↪  `is_type` targets.

#### Attributes to extract for `is_type`

None.

#### Examples

**Note:** These examples focus on the `is_type` relationship. Other supported relationships are
↪  omitted from the examples.

**Positive example:** Beta Systems developed Beta-1, a new industrial robot.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: [`Beta-1`]
  - **PRODUCT_TYPE**: [`industrial robot`]
  - **GPE**: []
  - **DATE**: []
- **is_type** relationships:
  - **is_type**
    - Source: `Beta-1`: **NAMED_PRODUCT**
    - Target: `industrial robot`: **PRODUCT_TYPE**
- Reason: The text directly identifies Beta-1 as an industrial robot.

**Negative example:** Beta-1 is produced by Beta Systems. Beta Systems produces industrial
↪  robots as well as cleaning robots.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: [`Beta-1`]
  - **PRODUCT_TYPE**: [`industrial robots`, `cleaning robots`]
  - **GPE**: []
  - **DATE**: []
- **is_type** relationships: None
- Reason: The text establishes that Beta Systems produces Beta-1 and produces both product
↪  types, but it does not establish which product type Beta-1 belongs to.

### 4. located_in

#### Definition

- A `located_in` relationship links an **ORG** to the **GPE** in which the **ORG** is located.

**Source:** **ORG** - the organisation whose location is specified.
**Target:** **GPE** - the geopolitical entity in which the source organisation is located.

Extract `located_in` when the text establishes that:
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
- the **ORG** has its headquarters, registered office, branch, factory, or another physical
↪  facility in the **GPE**
- the text identifies the **ORG**'s location through a geographic expression that directly
↪  modifies the **ORG**, and the corresponding **GPE** appears in the provided **GPE** list
  - Examples include `a French company`, `a Munich-based company`, and `Germany's Beta Systems`.
  - A geographic description qualifies only when it describes the **ORG** itself.
  - Descriptions of the **ORG**'s owners, founders, products, customers, subsidiaries, or
  ↪  business activities do not establish the **ORG**'s location.
  - Treat a geographic adjective only as evidence and use the corresponding provided **GPE** as
  ↪  the relationship target.

Do not extract `located_in` merely because the **ORG**:

- does business in the **GPE**, including selling or supplying a **NAMED_PRODUCT** or
↪  **PRODUCT_TYPE** or serving customers there, without separate textual evidence that
↪  establishes the **ORG**'s location
- attends or stages an event in the **GPE**

#### Attributes to extract for `located_in`

- `date_mention`: **DATE**[]. Apply General Rule 8.

#### Examples

**Note:** These examples focus on the `located_in` relationship. Other supported relationships
↪  are omitted from the examples.

**Positive example:** Beta Systems has been headquartered in Munich since 2020.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: []
  - **PRODUCT_TYPE**: []
  - **GPE**: [`Munich`]
  - **DATE**: [`2020`]
- **located_in** relationships:
  - **located_in**
    - Source: `Beta Systems`: **ORG**
    - Target: `Munich`: **GPE**
    - Attributes:
      - `date_mention`:
        - `2020`: **DATE**

**Negative example:** Beta Systems staged a one-night product showcase in Hamburg.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: []
  - **PRODUCT_TYPE**: []
  - **GPE**: [`Hamburg`]
  - **DATE**: []
- **located_in** relationships: None
- Reason: Staging a product showcase in Hamburg does not establish that Beta Systems is located
↪  there.

## Aliases

### Definition

- An alias is a text span that refers to the same underlying entity as another text span.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
- An alias group is a list of text spans that all refer to the same entity and have the same
↪  entity type.
- For each alias group, provide the entity type and the complete list of aliases.

**Note:** The examples below focus on aliases. Supported relationships are omitted.

### Alias Rule 1: Use Only Provided Entity Spans

- Build alias groups only from spans in the provided entity-type list.
- Every alias must exactly match a span in the list corresponding to its entity type.
- Do not create, expand, shorten, or relabel spans.

### Alias Rule 2: Require at Least Two Distinct Spans

Extract an alias group only when it contains at least two distinct aliases.

### Alias Rule 3: Require Equivalent References

Group spans into aliases only when they are interchangeable references established by the
↪  context.
- Entity names must refer to the same entity.
- **PRODUCT_TYPE** terms must additionally refer to the same product type at the same level of
↪  specificity.

**Positive example:** Beta Systems developed Beta-1, an industrial robot also known as B1. The
↪  company also developed an application that turns factory maintenance into guided tasks. The
↪  app alerts operators when Beta-1 needs service.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: [`Beta-1`, `B1`]
  - **PRODUCT_TYPE**: [`industrial robot`, `application`, `app`]
  - **GPE**: []
  - **DATE**: []
- Aliases:
  - **NAMED_PRODUCT**
    - `Beta-1`
    - `B1`
  - **PRODUCT_TYPE**
    - `application`
    - `app`
- Reason: `Beta-1` and `B1` explicitly refer to the same named product. `application` and `app`
↪  refer to the same product type at the same level of specificity.

**Negative example:** Gamma Aeronautics built the airplane that vanished over the ocean. Search
↪  crews later recovered the aircraft's flight recorder.
- Target entity: `gamma aeronautics`
- Input aliases: []
- Input entity lists:
  - **ORG**: [`Gamma Aeronautics`]
  - **NAMED_PRODUCT**: []
  - **PRODUCT_TYPE**: [`airplane`, `aircraft`, `flight recorder`]
  - **GPE**: []
  - **DATE**: []
- Aliases: None
- Reason: `airplane` is more specific than `aircraft`, so they do not refer to the same product
↪  type at the same level of specificity.

### Alias Rule 4: Do Not Output Overlapping Alias Groups

A text span may appear in at most one alias group. Therefore, alias groups must not overlap or
↪  share any text span.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
### Alias Rule 5: Capitalisation and Singular/Plural Variants

Treat capitalisation variants, and singular or plural variants of the same **PRODUCT_TYPE**, as
↪  aliases when the context clearly establishes that they refer to the same entity or product
↪  type. Include each distinct span exactly as provided.

**Positive example:** Beta Systems builds industrial robots. Each industrial robot is assembled
↪  in-house. Industrial robots from the company are used in car factories.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Beta Systems`]
  - **NAMED_PRODUCT**: []
  - **PRODUCT_TYPE**: [`industrial robots`, `industrial robot`, `Industrial robots`, `car`]
  - **GPE**: []
  - **DATE**: []
- Aliases:
  - **PRODUCT_TYPE**
    - `industrial robots`
    - `industrial robot`
    - `Industrial robots`
- Reason: All three text spans refer to the same product type. They differ only in
↪  capitalisation and whether they are singular or plural.

### Alias Rule 6: Extract Aliases Only for Relevant Entities

Extract aliases only for entities involved in a relationship that satisfies its
↪  relationship-specific definition and all applicable general annotation rules.

**Negative example:** Delta Industries released the new D-45, also known as Duck.
- Target entity: `beta systems`
- Input aliases: [`beta robotics`]
- Input entity lists:
  - **ORG**: [`Delta Industries`]
  - **NAMED_PRODUCT**: [`D-45`, `Duck`]
  - **PRODUCT_TYPE**: []
  - **GPE**: []
  - **DATE**: []
- Aliases: None
- Reason: Neither Delta Industries, D-45, nor Duck is involved in a relevant relationship with
↪  the target entity or a target-entity product.

### Alias Rule 7: Do Not Automatically Output Provided Aliases

The provided target-entity aliases should be used only to help identify the **target entity**.
↪  They should not be included in the output unless supported by the text.

## General Annotation Rules

### General Rule 1: Relationships Must Involve the Target Entity or Its Products

Extract a relationship only when at least one of the following applies:

- its source or target refers to the **target entity**
- a **NAMED_PRODUCT** or **PRODUCT_TYPE** used as its source, target, `named_product`, or
↪  `product_type` attribute is a **target-entity product**

Do not extract relationships unrelated to the **target entity** and its **target-entity
↪  products**.

### General Rule 2: Use Only Provided Entity Spans
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Every entity used as a relationship source, target, or attribute must exactly match a span in
↪  the provided list for its entity type. Do not create, expand, shorten, or relabel spans.

### General Rule 3: Use Only the Allowed Entity Types

For each relationship, use only the entity types specified for its source, target, and
↪  attributes.

### General Rule 4: Preserve Relationship Direction

Relationships are directed. For each relationship, match the entities properly to the
↪  **source** and **target** definitions used for that specific relationship type.
Do not reverse the relationship direction.

### General Rule 5: Consider Subtype Relationships and Complementarity for Product Types

When selecting **PRODUCT_TYPE** values for a relationship target or attribute, consider both
↪  subtype relationships and complementarity.

**Subtype relationships:**
- Treat one **PRODUCT_TYPE** as more specific than another only when it is clearly a strict
↪  subtype.
- Every instance of the more specific type must be an instance of the broader type, but not
↪  every instance of the broader type is an instance of the more specific type.
- For example, every road bike is a bicycle, but not every bicycle is a road bike. Therefore,
↪  `road bike` is a strict subtype of `bicycle`.

**Complementarity of product types:**
- **PRODUCT_TYPE** values are complementary when they describe the same product but neither is a
↪  strict subtype of the other.
- For example, a road bike is not necessarily a carbon bike, and a carbon bike is not
↪  necessarily a road bike. Neither term is therefore a subtype of the other. `road bike`
↪  describes the product's category or use, while `carbon bike` describes its material.
- Complementarity can also apply to product types that differ in granularity without forming a
↪  strict subtype relationship. For example, `robot` is a broad product category, whereas
↪  `surgical device` describes a more granular application category. Some products are both,
↪  but not every `robot` is a `surgical device`, and not every `surgical device` is a `robot`.

**Decision rule:**
- When one **PRODUCT_TYPE** is a strict subtype of another, select only the strict subtype.
- When multiple valid **PRODUCT_TYPE** values are complementary, select all of them.
- If a relationship attribute allows multiple **PRODUCT_TYPE** values, include all selected
↪  values in the same relationship instance.
- If the selected **PRODUCT_TYPE** values serve as the relationship target, create a separate
↪  relationship instance for each value.

### General Rule 6: Do Not Extract Duplicate Relationships

- Relationships are duplicates when they have the same **relationship type**, source entity,
↪  and target entity.
- Consolidate each set of duplicates by extracting one relationship and adding all supported
↪  attribute values to it.

### General Rule 7: Extract Only Existing or Past Relationships

- Extract only relationships that the text presents as existing now or having existed in the
↪  past.
- Do not extract relationships that are merely planned, proposed, possible, hypothetical,
↪  conditional, under consideration, or set to occur in the future.

### General Rule 8: Extract Only Dates Linked to the Relationship

For every relationship type that allows `date_mention`:
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
- include every **DATE** for which the text directly states or clearly supports that the
↪  specific relationship occurred, existed, or was active at that time
- the **DATE** does not need to identify the exact beginning or end of the relationship
- do not attach a publication, posting, announcement, or nearby date solely because it appears
↪  in the text
- every included **DATE** must be meaningfully linked to the specific relationship
- set `date_mention` to an empty list when no **DATE** qualifies

### General Rule 9: Extract Every Supported Relationship and Attribute

- Extract every relationship that satisfies the applicable relationship-specific definition and
↪  General Rules 1-8 and is either directly stated in the text or logically inferred and
↪  clearly supported by information in the text.
- Include every attribute supported to the same standard. When multiple relationship types are
↪  supported, extract each one.

## General Examples

**Note:** Unlike the focused examples above, these examples include all supported relationships
↪  and aliases.

**Example 1:** The bicycle maker Omega Cycling is introducing its newest gravel bike, the
↪  Super-G. The lightweight carbon bike costs €3,199.
- Target entity: `omega cycling`
- Input aliases: [`omegac`]
- Input entity lists:
  - **ORG**: [`Omega Cycling`]
  - **NAMED_PRODUCT**: [`Super-G`]
  - **PRODUCT_TYPE**: [`bicycle`, `gravel bike`, `carbon bike`]
  - **GPE**: []
  - **DATE**: []
- Relationships:
  - **makes**
    - Source: `Omega Cycling`: **ORG**
    - Target: `Super-G`: **NAMED_PRODUCT**
    - Attributes:
      - `date_mention`: []
  - **makes**
    - Source: `Omega Cycling`: **ORG**
    - Target: `gravel bike`: **PRODUCT_TYPE**
    - Attributes:
      - `date_mention`: []
  - **makes**
    - Source: `Omega Cycling`: **ORG**
    - Target: `carbon bike`: **PRODUCT_TYPE**
    - Attributes:
      - `date_mention`: []
  - **is_type**
    - Source: `Super-G`: **NAMED_PRODUCT**
    - Target: `gravel bike`: **PRODUCT_TYPE**
  - **is_type**
    - Source: `Super-G`: **NAMED_PRODUCT**
    - Target: `carbon bike`: **PRODUCT_TYPE**
- Aliases: None
- Do not extract:
  - `Omega Cycling makes bicycle`
  - `Super-G is_type bicycle`
- Reason: Omega Cycling is described as a bicycle maker, and Super-G is described as its newest
↪  bike. Together, these statements establish the `makes` relationships. Under General Rule 5,
↪  `gravel bike` and `carbon bike` are complementary because they describe the product's
↪  category and material, so both are extracted. The same rule excludes `bicycle` because it
↪  is broader and therefore redundant.
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
**Example 2:** Beta Systems manufactures the Beta-1 robot and supplies it to Delta Industries.
↪  Distributor Gamma also supplies Beta-1 robots to Epsilon Industries. Distributor Gamma
↪  manufactures Gamma-X sensors.
- Target entity: `beta systems`
- Input aliases: []
- Input entity lists:
  - **ORG**: [`Beta Systems`, `Delta Industries`, `Distributor Gamma`, `Epsilon Industries`]
  - **NAMED_PRODUCT**: [`Beta-1`, `Gamma-X`]
  - **PRODUCT_TYPE**: [`robot`, `robots`, `sensors`]
  - **GPE**: []
  - **DATE**: []
- Relationships:
  - **makes**
    - Source: `Beta Systems`: **ORG**
    - Target: `Beta-1`: **NAMED_PRODUCT**
    - Attributes:
      - `date_mention`: []
  - **makes**
    - Source: `Beta Systems`: **ORG**
    - Target: `robot`: **PRODUCT_TYPE**
    - Attributes:
      - `date_mention`: []
  - **is_type**
    - Source: `Beta-1`: **NAMED_PRODUCT**
    - Target: `robot`: **PRODUCT_TYPE**
  - **supplies_to**
    - Source: `Beta Systems`: **ORG**
    - Target: `Delta Industries`: **ORG**
    - Attributes:
      - `named_product`:
        - `Beta-1`: **NAMED_PRODUCT**
      - `product_type`:
        - `robot`: **PRODUCT_TYPE**
      - `date_mention`: []
  - **supplies_to**
    - Source: `Distributor Gamma`: **ORG**
    - Target: `Epsilon Industries`: **ORG**
    - Attributes:
      - `named_product`:
        - `Beta-1`: **NAMED_PRODUCT**
      - `product_type`:
        - `robots`: **PRODUCT_TYPE**
      - `date_mention`: []
- Aliases:
  - **PRODUCT_TYPE**
    - `robot`
    - `robots`
- Do not extract:
  - `Distributor Gamma makes Gamma-X`
  - `Distributor Gamma makes sensors`
  - `Gamma-X is_type sensors`
- Reason: The relationships involving Beta Systems or Beta-1 satisfy Rule 1. The second
↪  `supplies_to` relationship also satisfies Rule 1 because its `named_product` attribute is
↪  the target-entity product `Beta-1`. The Gamma-X relationships are unrelated to the target
↪  entity and its products. `robot` and `robots` are aliases because both refer to the same
↪  product type at the same level of specificity.

# Output Format

Answer in JSON using this schema:
{
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
// All relationships supported by the text and the relationship rules. Return an empty array
↪  when none are found.
relationships: [
  {
    relationship: "supplies_to",
    source: {
      entity_type: "ORG",
      span: string,
    },
    target: {
      entity_type: "ORG",
      span: string,
    },
    attributes: {
      // NAMED_PRODUCT mentions supplied in supplies_to. Null when none are stated.
      named_product: [
        {
          entity_type: "NAMED_PRODUCT",
          span: string,
        }
      ] or null,
      // PRODUCT_TYPE mentions supplied in supplies_to. Null when none are stated.
      product_type: [
        {
          entity_type: "PRODUCT_TYPE",
          span: string,
        }
      ] or null,
      // All DATE mentions explicitly connected with the supply relationship. Empty when none
      ↪  are stated.
      date_mention: [
        {
          entity_type: "DATE",
          span: string,
        }
      ],
    },
    reason: string,
  } or {
    relationship: "makes",
    source: {
      entity_type: "ORG",
      span: string,
    },
    target: {
      entity_type: "NAMED_PRODUCT",
      span: string,
    } or {
      entity_type: "PRODUCT_TYPE",
      span: string,
    },
    attributes: {
      // All DATE mentions explicitly connected with the relationship. Empty when none are
      ↪  stated.
      date_mention: [
        {
          entity_type: "DATE",
          span: string,
        }
      ],
    },
    reason: string,
  } or {
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
      relationship: "is_type",
      source: {
        entity_type: "NAMED_PRODUCT",
        span: string,
      },
      target: {
        entity_type: "PRODUCT_TYPE",
        span: string,
      },
      reason: string,
    } or {
      relationship: "located_in",
      source: {
        entity_type: "ORG",
        span: string,
      },
      target: {
        entity_type: "GPE",
        span: string,
      },
      attributes: {
        // All DATE mentions explicitly connected with the relationship. Empty when none are
        ↪  stated.
        date_mention: [
          {
            entity_type: "DATE",
            span: string,
          }
        ],
      },
      reason: string,
    }
  ],
  // Why the text supports none of the defined relationship types. Required when relationships
  ↪  is empty and null otherwise.
  no_relationship_reason: string or null,
  // Alias groups explicitly found in the text. Return an empty array when none are found.
  aliases: [
    {
      entity_type: "ORG" or "NAMED_PRODUCT" or "PRODUCT_TYPE" or "DATE" or "GPE",
      // The exact text spans that refer to the same entity.
      aliases: string[],
    }
  ],
}
```

I.10.2 Example User Prompt

```
Text:
The Ukrainian aircraft manufacturer Antonov expanded into drone production

It is planning to build reconnaissance drones

Translated: false
Target entity: antonov
Input aliases: ["антонов"]
Input entity lists:
- ORG: ["Antonov"]
- NAMED_PRODUCT: []
- PRODUCT_TYPE: ["drone","reconnaissance drones"]
- GPE: []
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
- DATE: []
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.