DATA ENGINEERING AND ANALYTICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

A Unified Approach to Crosslingual News Similarity Evaluation: from Crowdsourcing to Large Language Models Optimization

Sukhodolskaya Evgeniya

DATA ENGINEERING AND ANALYTICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

A Unified Approach to Crosslingual News Similarity Evaluation: from Crowdsourcing to Large Language Models Optimization

Ein einheitlicher Ansatz zur Bewertung von Ähnlichkeiten in mehrsprachigen Nachrichten: Von Crowdsourcing zur Optimierung von Large Language Models

Author:	Sukhodolskaya Evgeniya
Supervisor:	Prof. Dr. Georg Groh
Advisor:	Daryna Dementieva
Submission Date:	15.09.2024

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 15.09.2024 Sukhodolskaya Evgeniya

Acknowledgments

In this acknowledgement part of my thesis, I extend my appreciation to all who have been part of this journey. I thank my advisor, a postdoctoral researcher at the Technical University of Munich, Daryna Dementieva, my role model as a woman NLP researcher, for her unbreakable motivation to push Ukrainian NLP forward. It requires a special mention of the importance of having a female role model in professional and research STEM fields. I thank Toloka crowdsourcing company for recognizing the need to contribute to open-sourced research and supporting it with grants. I thank my mother, Tatiana, the most intelligent person I know, without who I'd never end up completing my master's degree in Germany and becoming a high-valued expert in IT. Lastly, I thank my partner, Mario, who kept me sane and fed throughout these six months of hard work.

Abstract

As news volumes and the spread of misinformation rise, the task of measuring news semantic similarity becomes critical for understanding global media coverage. While existing Natural Language Processing (NLP) solutions offer tools for measuring semantic similarity, linguistic diversity and high subjectivity make the task particularly difficult in the news domain. The lack of data for low-resourced languages further complicates the task.

To bridge this gap, we developed a reusable, scalable crowdsourcing pipeline for collecting a high-quality multilingual semantic similarity dataset of news articles in Ukrainian, Russian, Polish, and English. The dataset includes 500 news pairs labelled for similarity using the 4W method (What, Where, When, and Who), along with textual justifications for the chosen similarity level.

Our research benchmarks several modern transformer-based models, revealing that even state-of-the-art models struggle with news semantic similarity measurement, particularly in low-resourced languages. However, we identified promising candidates for future down-streaming with the collected dataset.

Despite limitations, such as excluding sensitive news topics and resource constraints that prevented the evaluation of closed-source models, this work provides valuable contributions to the NLP field. It offers a framework for future research, particularly expanding multilingual news semantic similarity datasets for low-resourced languages.

Kurzfassung

Da das Nachrichtenaufkommen und die Verbreitung von Fehlinformationen zunehmen, wird die Messung der semantischen Ähnlichkeit von Nachrichten für das Verständnis der globalen Medienberichterstattung immer wichtiger. Zwar bieten bestehende Lösungen für die Verarbeitung natürlicher Sprache (Natural Language Processing, NLP) Werkzeuge zur Messung der semantischen Ähnlichkeit, doch die sprachliche Vielfalt und die hohe Subjektivität erschweren diese Aufgabe im Nachrichtenbereich besonders. Der Mangel an Daten für Sprachen mit geringen Ressourcen verstärkt diese Herausforderung zusätzlich.

Um diese Lücke zu schließen, haben wir eine wiederverwendbare und skalierbare Crowdsourcing-Pipeline entwickelt, die einen hochwertigen, mehrsprachigen Datensatz zur semantischen Ähnlichkeit von Nachrichtenartikeln in den Sprachen Ukrainisch, Russisch, Polnisch und Englisch sammelt. Der Datensatz enthält 500 Nachrichtenpaare, die anhand der 4W-Methode (Was, Wo, Wann und Wer) auf Ähnlichkeit geprüft wurden, begleitet von textlichen Begründungen für den gewählten Ähnlichkeitsgrad.

Unsere Forschung testet mehrere moderne, auf Transformatoren basierende Modelle und zeigt, dass selbst die fortschrittlichsten Modelle Schwierigkeiten bei der Messung der semantischen Ähnlichkeit von Nachrichten haben, insbesondere in Sprachen mit geringen Ressourcen. Dennoch haben wir mit dem gesammelten Datensatz vielversprechende Kandidaten für zukünftige Downstream-Anwendungen identifiziert.

Trotz einiger Einschränkungen, wie dem Ausschluss sensibler Nachrichtenthemen und Ressourcenbeschränkungen, die die Evaluierung von Closed-Source-Modellen verhinderten, leistet diese Arbeit einen wertvollen Beitrag zum NLP-Bereich. Sie bietet einen Rahmen für zukünftige Forschungen, insbesondere zur Erweiterung mehrsprachiger Datensätze zur semantischen Ähnlichkeit von Nachrichten in Sprachen mit geringen Ressourcen.

Contents

1. Introduction

1.1. Motivation

In today's world, the volume of news we face daily continues to grow, and consequently, grows the spread of misinformation. With the extensive amount of news articles published daily, an efficient method for measuring the similarity between news articles is crucial for clustering and identifying coverage of the same events across different media outlets, fake news and propaganda. Furthermore, quantifying the similarity between news articles can help track which stories dominate the media landscape and monitor how news stories spread across different media ecosystems over time [1].

Although numerous natural language processing (NLP) solutions exist for semantic similarity measurement, the news semantic similarity domain presents unique challenges. Precisely measuring news similarity requires extensive world knowledge, such as understanding geopolitical events, historical context, and cultural references, and gets even more complicated due to the diverse writing styles, tones, and terminology used in different news sources. The task becomes even more complicated when considering multilingual news similarity measurement, where differences between languages add additional layers of context. However, being able to measure news similarity across languages opens up significant opportunities to fight misinformation and provide a more comprehensive understanding of global news coverage.

In recent years, the shift toward a data-centric approach has highlighted the importance of high-quality datasets for the progress of modern NLP. Large language models (LLMs) have demonstrated surprising emergent abilities, and their full potential cannot be measured using standard benchmarks alone. Therefore, providing high-quality datasets to the NLP research community for benchmarking is essential to advance the multilingual news similarity measurement field. However, creating scalable pipelines for collecting such datasets, especially for low-resource languages, remains challenging. The task of news semantic similarity is difficult to formalize, and it can be highly subjective, which makes creating a high-quality dataset in this area seem impossible without access to domain expert groups.

Moreover, high-quality datasets are not only necessary for evaluation but are also crucial components for creating new performant NLP solutions, particularly in domains where smaller, fine-tuned models may outperform their larger counterparts, which might be the case in the news semantic similarity in low-resourced languages. Access to high-quality datasets for fine-tuning can lead to significant advancements for the open research community, which often lacks the resources to build massive generative models.

The Ukrainian language, which is considered low- to mid-resource depending on the estimation [2], faces challenges due to the scarcity of NLP solutions around it. In light of the Russian-Ukrainian war of 2022, the need for precise news semantic similarity measurement

in Ukrainian, both in monolingual and multilingual contexts, is high. To address this, there is a need for a high-quality multilingual news semantic similarity dataset in Ukrainian. This work aims to bridge this gap.

1.2. Scope of Work

This work addresses the following research questions:

RQ1 What is the current state of research in NLP regarding news semantic similarity measurement, particularly in multilingual settings?

RQ2 How can a scalable, high-quality crowdsourcing solution be designed to collect multilingual datasets in low-resourced languages for the task of multilingual news semantic similarity measurement?

RQ3 How do modern LLMs, including encoder-based and decoder-only ones, perform when applied to multilingual news semantic similarity measurement, especially in low-resourced languages?

The first question is explored in **Chapter 2**, which reviews existing research in the field of news semantic similarity measurement, focusing on challenges introduced by multilingual and low-resourced language settings. **Chapter 3** addresses the second research question by presenting extensive guidelines for the development of a pipeline for crowdsourcing a multilingual news semantic similarity dataset covering Ukrainian, Russian, Polish, and English languages. This chapter also introduces an automated LLM-based quality control system with Human-in-the-Loop (HITL) designed to ensure the scalability of the crowdsourcing process while preserving the high quality of results. Finally, **Chapter 4** answers the third research question by benchmarking modern encoder-only, encoder-decoder, and decoder-only models on the collected dataset, providing an in-depth analysis of their performance on multilingual news semantic similarity measurement.

2. Literature Review

In **Section 2.1**, we review existing research on measuring textual semantic similarity, including a general overview of the field and its key approaches and models, including modern Large Language Models. We also discuss relevant datasets used for semantic similarity tasks. **Section 2.2** focuses on recent developments in multilingual news semantic similarity, particularly the SemEval 2022 task "Multilingual News Article Similarity" [3]. We explore the datasets provided for this task and highlight the most successful solutions from the competition.

2.1. Background

2.1.1. Textual Semantic Similarity

Semantic Textual Similarity is a crucial task in Natural Language Processing (NLP), referring to measuring how much meaning is shared between two text snippets, considering the semantic relationships between words. This task is particularly challenging since language units, especially words, often have multiple meanings [4]. Textual Semantic Similarity plays a significant role in various applications such as information retrieval, automatic question answering, machine translation, dialogue systems, and document matching.

Over the past thirty years, various techniques for measuring semantic similarity have been developed [5]. These methods can be classified in various ways, including by the unit of text (e.g., word, sentence, paragraph, document), the approach used to measure similarity (e.g., corpus-based, knowledge-based, hybrid methods), context-awareness, and the similarity metrics applied. For this discussion, the focus will be on the classification most commonly recognised within the NLP community, which is based on the approaches used [6].

Before exploring semantic methods, it's worth briefly mentioning some widely used textual representation techniques that, while not inherently semantic, are frequently employed in textual similarity measurements. These include Bag-of-Words (BoW) [7], Term Frequency-Inverse Document Frequency (TF-IDF) [8], and BM25 [9]. Although these methods do not account for the context or meaning of words, they serve as useful benchmarks, are straightforward to compute, and are language-agnostic. Now, let's explore the semantic methods in more detail.

Knowledge-Based Semantic Similarity Methods

Knowledge-based semantic similarity methods measure the similarity between terms in texts using information from knowledge sources such as ontologies, lexical databases, thesauri,

and dictionaries. These methods effectively capture the deep semantic meaning of textual units because the terms and their relationships are comprehensively described within these knowledge sources [10]. Commonly used knowledge sources include WordNet [11], Wikipedia [12], and BabelNet [13].

These methods are often linked to graph theory, particularly in approaches like *edge-counting*, where the underlying ontology is considered a graph with words connected taxonomically. The similarity between two terms is measured by counting the edges (connections) between terms within this graph [14]. Alternatively, *feature-based* methods calculate similarity based on the properties of the words, and *information content-based* methods assess similarity based on the specificity of the terms [15].

Knowledge-based semantic similarity methods are not as computationally intensive as other approaches and can address the common challenge of ambiguity in textual similarity measurements. These methods effectively handle synonyms, idioms, and colloquial phrases, although they may sometimes overlook semantic relatedness [4]. However, the accuracy of these methods is highly dependent on the underlying knowledge source, which can pose challenges for low-resource languages, specific domains, and maintaining up-to-date information.

Corpus-Based Semantic Similarity Methods

Corpus-based semantic similarity methods are based on the "distributional hypothesis," which states that the meaning of a word can be inferred from the context in which it appears—essentially, "You shall know a word by the company it keeps." [CIT] These methods assess the semantic similarity between terms by analysing information from large text corpora, focusing on the surrounding or co-occurring words, often referred to as the *context* [16, 4].

Typically, these methods represent words, sentences, paragraphs, or entire documents as vectors in a space defined by the corpus vocabulary. This representation is known as an *embedding*. Various metrics are used to estimate the similarity between these vectors, including Cosine Similarity, Dot Product, Manhattan Distance, Euclidean Distance, Jaccard Index, and others, with *Cosine Similarity* being the most widely used method to date [6].

Some of the most prominent models developed in the last decade for computing dense embedding representations of text include:

- **word2vec**: A simple three-layer neural network model developed using the Google News dataset [17]. Later extended to a model producing document embeddings (doc2vec) [18].
- **GloVe**: This model creates word embeddings based on a global word co-occurrence matrix derived from the underlying corpus [19].
- **fastText**: This model builds word vectors using Skip-gram models where each word is represented as a collection of character n-grams, which accounts for the morphological structure of words [20].

- **BERT (Bidirectional Encoder Representations from Transformers)**: A transformer-based model that produces attention-based word embeddings [21, 22]. BERT has been extended to create sentence embeddings (Sentence-BERT) [23].

Corpus-based semantic similarity methods can be categorised into various approaches, including:

- *Latent Semantic Analysis (LSA)* [24]
- *Hyperspace Analogue to Language (HAL)* [25]
- *Explicit Semantic Analysis (ESA)* [26]
- *Word-alignment Models* [27]
- *Latent Dirichlet Allocation (LDA)* [28]
- *Normalized Google Distance* [29]
- *Dependency-based Models* [30]
- *Kernel-based Models* [31]
- *Word-attention Models* [32]

In the taxonomy used in [6], Deep Neural Networks (DNN)–based methods are classified separately since they are directly used to estimate the similarity between word embeddings. However, since deep neural network-based methods also exploit word embeddings built using large corpora, we decided to keep them within this class of approaches. DNN methods, in addition to the already mentioned models for constructing embeddings, are also based on Convolutional Neural Networks (CNN), variations of Long Short Term Memory architecture (LSTM), or Transformer-based architectures [33, 34, 35]. After Devlin et al. used the transformer model to generate BERT word embeddings in 2019 [22], NLP experienced a boom of models based on encoder-decoder architecture, which left BERT behind. Among these models are XLNet [36], RoBERTa [37], and T5 [38]. The success of T5 models, demonstrating the power law in scaling neural networks [39], was a precursor to the current prominence of Large Language Models (LLMs) in NLP.

Unlike knowledge-based systems, corpus-based systems are
language- and domain-independent. However, corpus-based methods do not account for the actual meaning of words. One of the major challenges when deploying word embeddings to measure similarity is *Meaning Conflation Deficiency*, where there is no differentiation between the various meanings of a word. This deficiency can introduce noise into the semantic space by bringing irrelevant words closer together. For example, the word "nail" could bring "manicure" and "hammer" together as related words. [40].

It might be that Large Language Models, which are also corpus-based and which we will briefly describe in the next section, mitigate the problem of „Meaning Conflation Deficiency“

to a certain extent. Nevertheless, Large Language Models exemplify another challenge of corpus-based semantic similarity methods: computational complexity. [40]

Any Machine Learning approach would not succeed without the right data for training, fine-tuning, evaluation, and benchmarking [41, 42]. Before moving to the next section, we want to briefly mention the most vital group of datasets for Semantic Textual Similarity in Natural Language Processing.

These datasets arise from the existence of SemEval[1]—a series of international workshops that have focused on semantic analysis and understanding since 2007. SemEval has helped create high-quality annotated datasets for increasingly challenging problems in Natural Language Semantics. Thanks to SemEval, we have the *STS-Benchmark*—the most famous primarily English benchmark for Semantic Textual Similarity, resulting from SemEval 2017 [43]. The STS-Benchmark consists of sentence pairs from various domains, labelled with a similarity score between 0 and 5, where 0 indicates complete dissimilarity, and 5 indicates complete semantic equivalence. It was annotated with the help of the crowdsourcing platform Amazon Mechanical Turk[2]. This benchmark is part of the well-known General Language Understanding Evaluation (GLUE) [44] and Massive Text Embedding Benchmark (MTEB) [45] benchmarks, along with other STS datasets collected through SemEval. This work is significantly influenced by SemEval 2022, which will be reviewed in further sections.

2.1.2. Large Language Models and Semantic Textual Similarity

Model Type	**Training**	**Model Type**	**Pretrain Task**
Encoder-Decoder or Encoder-only (BERT-style)	Masked Language Models	Discriminative	Predict masked words
Decoder-only (GPT-style)	Autoregressive Language Models	Generative	Predict the next word

Table 2.1.: Transformers Architectures

The combination of transformer architecture with the Power Law effect — where scaling up language models significantly improves few-shot and even zero-shot performance — has greatly influenced modern NLP. Large language models (LLMs) refer to transformer-based models that have hundreds of billions of parameters, trained on vast amounts of text data [46].

Based on the encoder-decoder architecture, three main approaches have emerged for model development: *encoder-only*, *encoder-decoder*, and *decoder-only*. The key differences of these approaches, evolution of which is shown in Figure 2.1, are summarized in Table 2.1. Following the initial rise in popularity with BERT, encoder-only models have seen a gradual decline. In contrast, decoder-only models have become a focus of the field, especially with the development of GPT-3 model [42]. Released in 2020 with 175 billion parameters, GPT-3

[1]https://aclanthology.org/venues/semeval/
[2]https://www.mturk.com/

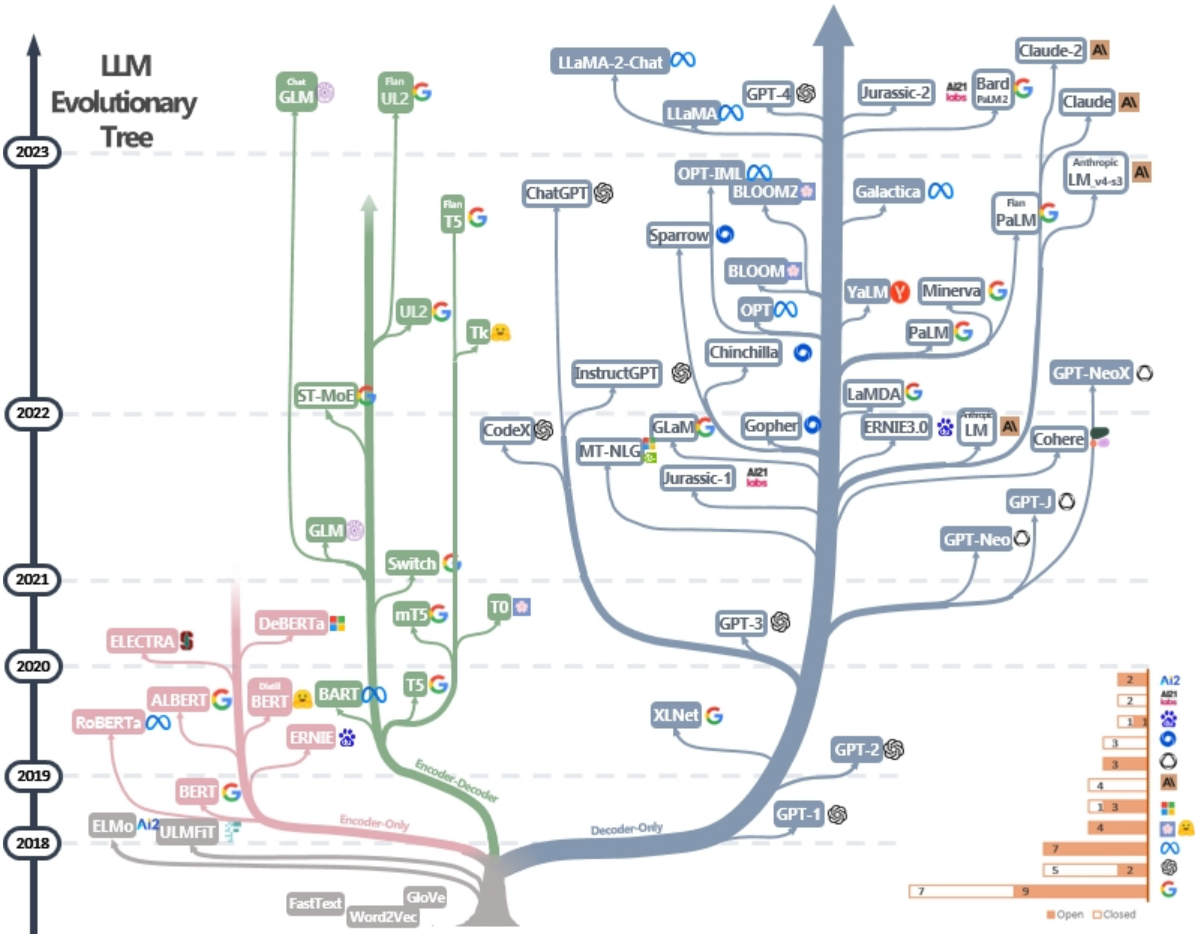


Figure 2.1.: The evolutionary tree of modern LLMs. Source: Yang et al. [42]

introduced the concept of in-context learning (ICL), allowing LLMs to understand tasks through natural language instructions [47].

Some of the leading modern LLMs include *GPT-4* [48], *Mistral 7B* [49], models from the *Llama* family [50], *Command-R* & *Aya* [51] from Cohere, Anthropic's *Claude* [52], and Google DeepMind's *Gemini* [53]. These models are primarily developed within the industry, and for most, key details about their training processes—such as data collection and cleaning methods—remain undisclosed [46, 54].

The most impressive achievements of Large Language Models lie in their emergent abilities: in-context learning, instruction following, and step-by-step reasoning [55]. However, their performance on specific tasks and low-resourced languages remains under-explored and is heavily influenced by the data used during pretraining, which is rarely fully disclosed. Common datasets for training large language models include *Common Crawl* [56], *WebText* [57], *BookCorpus* [58], *BigQuery* [59], and *Wikipedia-based* sources [60]. These datasets contain books, Reddit discussions, code, scientific articles, and, notably, for our purposes, news articles. For instance, they include *CC-News* and *RealNews*, the majority of which are, however, in English [46].

When evaluating the performance of Large Language Models on Semantic Textual Similarity tasks, research literature has limited results. It is noted that fine-tuned models—smaller language models pretrained and then further optimized on a task-specific dataset—generally outperform LLMs in traditional NLU tasks, especially when these tasks come with rich, well-annotated data and few out-of-distribution examples in the test sets [42]. However, LLMs tend to be more effective when working with limited annotated data, except when it comes to tasks involving extremely low-resource languages. Both LLMs and fine-tuned models can be suitable when abundant annotated data is available, depending on the specific task requirements. While LLMs benchmarked on the *MTEB benchmark* are also evaluated on the STS task, this evaluation does not extend to document-to-document similarity [46]. The study by Gatto et al. [61] indicates that generative LLMs significantly outperform existing encoder-based STS models when assessing the Semantic Similarity between texts with complex semantic relationships dependent on world knowledge. However, the performance of LLMs in low-resourced languages or in domain-specific tasks remains uncertain due to the scarcity of human-annotated data [62].

2.2. Related Work

Formalizing the task of measuring Semantic Similarity in news articles is challenging due to the inherent differences in length, tone, writing style, use of key terms, slang, and abbreviations, even within the same language. Semantic textual similarity methods generally quantify the degree to which two arbitrary documents are "similar," often without providing a precise definition of what this similarity entails or only doing so in very general terms, which is especially challenging to verify in the domain of news articles. [1].

News article similarity is related to the comparison of narratives, which requires an understanding of both structure and content to accurately assess similarity [1]. For over a century,

the checklist known as *5W 1H* has been a fundamental tool in journalism for summarizing the key points of a story. This approach involves answering five questions starting with "W" (What, Where, When, Who, Which) and one question starting with "H" (How) [63]. The semantic similarity of news articles intuitively resembles the task of comparing these essential points, although extracting these elements is a challenging task in itself.

In the research literature, approaches to measuring news semantic similarity are often knowledge-based, such as those utilizing WordNet [64]. However, maintaining up-to-date ontologies for news terms worldwide is practically impossible due to the fast-paced and ever-evolving nature of the field [65]. Regarding corpus-based methods, two primary approaches are used for semantic news comparison: *supervised* and *unsupervised*. Supervised methods face challenges due to the scarcity of training data, which is extremely difficult to collect, and the issues of granularity and explainability of similarity values. [66]. Unsupervised methods, particularly embedding-based approaches, often rely on older techniques like Word2Vec and FastText [67], with limited application of the latest state-of-the-art models (SOTA) in the field. However, even these SOTA models have been reported to struggle when dealing with low-resourced languages. According to recent research, Ukrainian is still considered a low-resourced language [2]. Expanding the focus of news semantic similarity to include low-resourced languages is crucial, especially to combat the spread of fake news in recent years [68].

High-quality news similarity datasets are essential for strengthening supervised approaches to measuring news Semantic Similarity. While they exist in limited quantities, such as the "Czech News Dataset for Semantic Textual Similarity" [69], cross-lingual news similarity measurement datasets were largely absent until the SemEval 2022 competition.

2.2.1. Multilingual news article similarity

The "SemEval 2022, Task 8" introduced a competition titled "Multilingual News Article Similarity."[3] This task aimed to develop systems that identify multilingual news articles containing similar information, with a focus on real-world events covered in the news, rather than the style of writing, political spin, tone, or other subjective elements imposed by the media. The goal was to enable the development of applications that can cluster news articles and track the similarity of news coverage across different outlets or regions [1].

Participants in this task were provided with a large collection of news articles, including 4,918 pairs with golden similarity labels. They were tasked with estimating the overall similarity of 4,902 news article pairs. The challenge was empathized by the language diversity of the task: the training data included 8 language combinations, while the evaluation dataset had 18 language combinations, with three languages not appearing in the training data at all.

To facilitate the competition, the authors created a high-quality dataset by annotating pairs of news articles from January 1, 2020, to June 30, 2020, sourced from Media Cloud [70]. They developed an annotation process centered around seven dimensions of similarity, as outlined in Table 2.2, covering 10 different languages: English, Spanish, Russian, German, French,

[3]https://competitions.codalab.org/competitions/33835

GEO	How similar is the geographic focus (places, cities, countries, etc.) of the two articles?
ENT	How similar are the named entities (e.g., people, companies, organizations, products, named living beings), excluding previously considered locations appearing in the two articles?
TIME	Are the two articles relevant to similar time periods or describing similar time periods?
NAR	How similar are the narrative schemas presented in the two articles?
OVERALL	Overall, are the two articles covering the same substantive news story? (excluding style, framing, and tone)
STYLE	Do the articles have similar writing styles?
TONE	Do the articles have similar tones?

Table 2.2.: Questions for assessing the similarity between two articles.

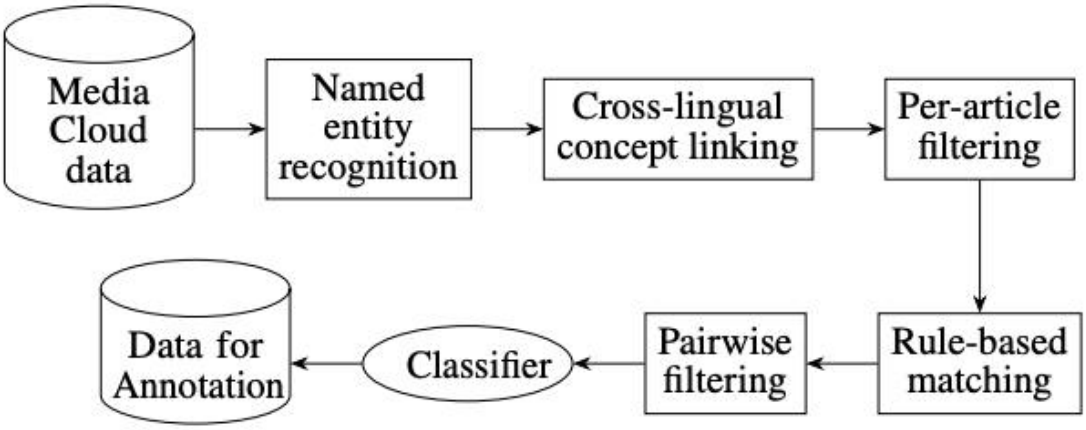


Figure 2.2.: SemEval2022 dataset preparation pipeline. Source: Chen et al. [1]

languages	annotations	mean(OVERALL)
en	5,189	2.92
de	2,166	2.56
es	955	2.40
zh	866	2.24
de-en	863	3.20
tr	817	2.79
pl	584	2.36
ar	572	2.41
es-en	504	2.79
it	411	2.65
es-it	320	2.29
ru	289	2.78
zh-en	253	3.04
fr	184	2.39
de-fr	116	1.88
pl-en	77	2.38
de-pl	35	1.69
fr-pl	11	1.91

Table 2.3.: Data annotated for the SemEval2022 competition. Source: Chen et al. [1]

Arabic, Italian, Turkish, Polish, and Mandarin Chinese. News articles were filtered, matched, and sampled according to the process depicted in Figure 2.2.

Annotators responded to each question using a four-point Likert scale with the options "Very Dissimilar," "Somewhat Dissimilar," "Somewhat Similar," and "Very Similar," following detailed annotation guidelines [3]. The total number of annotations used for the training and evaluation data, as well as the mean "OVERALL" label for news similarity (with higher values indicating greater dissimilarity) across the 10 languages and their combinations, is presented in Table 2.3.

The top solutions from the competition are summarized in Table 2.4. Other participants experimented with fine-tuning, combining, and stacking various transformer models, including DistilBERT, BERT, RoBERTa, XLM, and INFOXLM [75], [76], [77].

Overall, systems that utilized multiple sections of the news article and those that fine-tuned or otherwise trained embedding models generally outperformed those that did not. There were significant variations in performance across different languages, leading the competition authors to conclude that further research is necessary to develop multilingual systems capable of handling diverse language combinations. This work aims to address that gap, particularly given that, to our knowledge, no datasets or evaluated approaches currently exist for Multilingual News Similarity focused on the Ukrainian language.

Aspect	**Xu et al.** [71]	**Singh et al.** [72]
Place in the Competition	First	Second
Model	XLM-RoBERTa	Siamese LaBSE [73]
Data Augmentation	Back-Translation; Translate-Train	BM25 Sampling; Machine Translation to English
Text Truncation Strategy	Head-tail	Head
Similarity Measurement	Mean Squared Error on XLM-Roberta [CLS] output token	Mean Squared Error on Cosine Similarity between Extracted Named Entities and Siamese LaBSe output
Key Insight	R-Drop [74]; Importance of data quality in cross-lingual tasks	Among the Siamese Transformers, LaBSE works best

Table 2.4.: Top Solutions of the SemEval2022 competition

3. Dataset Collection

Based on our review of related research works, we found that the latest NLP tools cannot be expected to provide high-quality results for estimating semantic similarity between news articles. Therefore, there is a need for a crowdsourcing solution as a dataset-gathering method to bridge this gap.

We chose to build our pipeline on the experience of the SemEval2022 organizers, adapting it to a broader scalable scenario. Instead of relying on academic experts for labelling; we decided to collect data through a crowdsourcing platform. While this approach offers greater scalability, it also presents significant challenges in maintaining the quality of labelling and preventing fraud. Creating concise instruction, a user-friendly labelling interface, and a well-decomposed task structure is crucial.

That led us to limit the labelling options for news similarity to "Similar," "Somewhat related," and "Different" instead of using the four options from the SemEval task and to decompose the task according to the 4W model, which was described in **Chapter 2**. Additionally, to enhance the explainability and increase the value of the collected data, we added an extra labelling requirement: annotators are asked to provide an explanation for their similarity rating estimation between the news events in the paired articles. These explanations, ideally summarizing the main events in the news, could serve as a valuable resource for fine-tuning NLP models.

3.1. Gathering Data for labeling

To reflect the real-world distribution of news articles — considering variations in length, tone, style, and focus — we needed to devise a method to scrape data from actual news sources. The pipeline we developed can be seen in Figure 3.1. We will go through it step by step.

3.1.1. Scraping

Finding a balanced amount of news pairs with different levels of similarity is a challenging task. To tackle this, we decided to reuse the labels provided in the SemEval2022 Task. We equally sampled four groups of news pairs labelled as "Very Similar," "Somewhat Similar," "Somewhat Dissimilar," and "Very Dissimilar." These pairs included news articles in Russian, Polish, and English — languages most likely to cover news events affecting Ukraine.

To identify similar news events for each pair, we had to find a concise representation of the article's main events. We quickly figured out that not all news articles have representative titles. Therefore, we decided to extract the top 10 keywords from each article and base our search on them. To extract the keywords, we needed to extract each article's full text. For

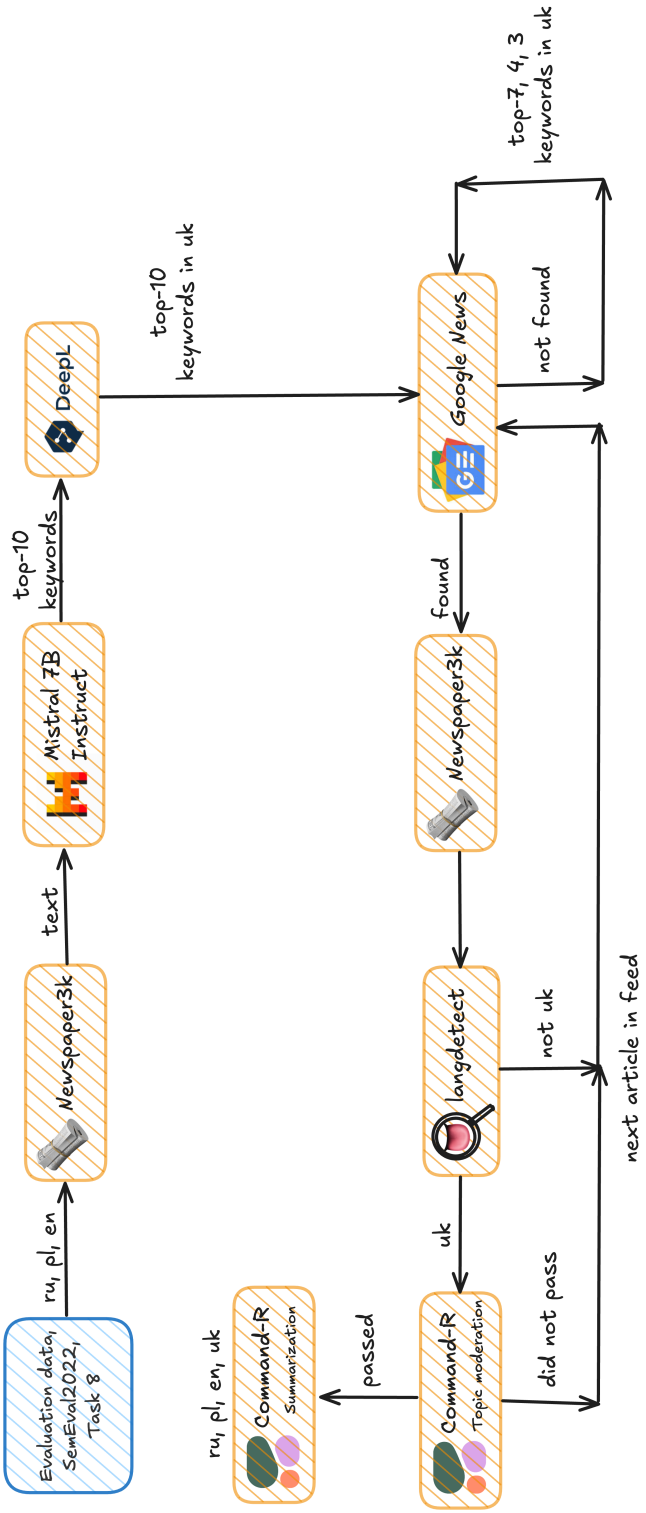


Figure 3.1.: Scraping pipeline

this purpose, we used the python **Newspaper3k** library[1]. Although this library also offers extractive summarization and keyword extraction, we opted for abstractive methods to achieve a better semantic representation of an article. We selected the **Mistral 7B Instruct** model from HuggingFace[2] for this task since it is an open-source, accessible through the HuggingFace API[3] model, which has empirically demonstrated high-quality keywords generation. To improve the quality of the results, we used all the prompting strategies recommended for this model[4]. The one-shot prompt we used is as follows:

Prompt, Mistral 7B Instruct, Keywords

```
<s>[INST] Extract top-10 keywords from a news article.
Order them from the most informative to the least,
in the following format:
1. <top-1 keyword>;
2. <top-2 keyword>;
... ;
10. <top-10 keyword>;
The news article:
"The weather will suddenly change in Kyiv on 13 March;
strong winds will rise in the capital of Ukraine.
This was reported by the press service of the State Emergency Service
of Ukraine in Kyiv with reference to the Ukrhydrometcenter.
In the city, the 1st level of danger is announced - yellow.
‘‘In Kyiv in the next hour and until the end of the day on 13 March.
we will have wind gusts of 15-18 m/s strength,’’ - stated the message.
The weather will change dramatically in Kyiv. As reported by OBOZREVATEL,
in Kyiv, in March, real spring will come -
it will get warmer, and instead of snow, it will only rain." [/INST]
1. weather;
2. Ukraine;
3. danger;
4. Kyiv;
5. wind;
6. Ukrhydrometcenter;
7. March;
8. change;
9. spring;
10. strong; </s> [INST]
```

[1]https://newspaper.readthedocs.io/en/latest/
[2]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[3]https://huggingface.co/docs/api-inference
[4]https://www.promptingguide.ai/models/mistral-7b

```
Extract 10 keywords from a news article.
The news article: "{NEWS ARTICLE}"
Order them from the most informative to the least,
in the following format:
1. <top-1 keyword>;
2. <top-2 keyword>;
... ;
10. <top-10 keyword>; [/INST]"
```

After obtaining the keywords, we translated them into Ukrainian using **DeepL**[5]. While there are various news aggregators available, we used the **GoogleNews**[6] Python library to scrape news from Google News, being the most widely used one. Initially, we tried using Google directly as a source for news articles. However, the results were too noisy, including unrelated materials like books and blogs, which were challenging to filter out. For our search queries, we concatenated the top 10 keywords with spaces. If no relevant articles were found, we gradually reduced the number of keywords to seven, four, and finally three, to increase the chances of retrieval.

Two main challenges prevented us from simply scraping news articles directly without pre-filtering. First, even though Google News allows region and language selection, the results are not always fully aligned with these filters. To address this, we utilized the Python library **langdetect**[7] to ensure that the text we scraped using the Newspaper3k library was in Ukrainian. Second, many crowdsourcing platforms implement moderation mechanisms to ensure compliance with their guidelines. For our project, we chose **Toloka** platform, which has a set of moderation rules[8] that require pre-filtering of the sensitive content. Since it is almost impossible to avoid sensitive topics when dealing with news, we decided to exclude news articles related to the 2022 Russian-Ukrainian War after consulting with Toloka's support team. To automate the detection and exclusion of news articles regarding this conflict, we used the **Command R**[9] model from Cohere.

Command R is a language model designed to improve the accuracy and relevance of text classification by incorporating external knowledge through retrieval. It is built on a large-scale transformer architecture and supports over 100 languages, which makes it suitable for multilingual tasks. We found that Command R performed better at detecting sensitive topics than other open-source models like Mistral 7B. To ensure the best results, we followed Cohere's prompting guidelines[10].

Our final dataset consisted of pairs of news articles, where each pair contained either two articles for which we successfully found Ukrainian doubles that passed our automated pre-filtering or one article that had found a match in Ukrainian. It is important to note that

[5]https://www.deepl.com/
[6]https://pypi.org/project/GoogleNews/
[7]https://pypi.org/project/langdetect/
[8]https://toloka.ai/docs/guide/unwanted/
[9]https://docs.cohere.com/docs/command-r
[10]https://docs.cohere.com/docs/crafting-effective-prompt

keyword-based searches can't guarantee identical news to be found, especially when we had to reduce the number of keywords for broader retrieval.

3.1.2. Data Preparation

News articles can vary significantly in length, often with long and detailed texts. It is challenging to ensure that annotators on crowdsourcing platforms thoroughly read through entire articles. As attention focus tends to decrease in large-scale crowdsourcing projects, we had to take steps to preserve annotation quality. To address this, we had two options: either provide guiding questions to help annotators focus on specific sections of the article or generate concise, information-dense summaries so annotators could selectively check details in the full text.

Summarization is a well-researched task in Natural Language Processing. With the recent rise of LLMs, numerous models have demonstrated the ability to produce high-quality abstractive summaries. Given this, we decided to use summarization to assist annotators, structuring the summaries in the same format as we decomposed the similarity measurement objective. Specifically, the summaries focused on specifying key aspects such as WHAT happened in the news article, WHEN and WHERE the events occurred, and WHO the main actors involved were.

For this task, we opted for the **Command R** model from Cohere, which we also used for filtering news articles due to its empirically noticeable performance. The prompt we used for generating summaries was as follows:

Prompt, Command R Cohere, Summarization, English

```
## Instructions

    Summarize a news article focusing on four main points:
    1. WHO is the main/secondary actor in the news article?
    2. WHEN did the situation described in the news article occur?
    3. WHERE did the situation described in the news article take place?
    4. WHAT happened in the news article?
    (200-250 words summarization of the news article)

    ## Example output
    WHO: <answer to the 1st point of the Instructions>
    WHEN: <answer to the 2nd point of the Instructions>
    WHERE: <answer to the 3rd point of the Instructions>
    WHAT: <answer to the 4th point of the Instructions
    (200-250 words summary of the news article)>.

    ## News Article
```

```
{NEWS ARTICLE}
```

Command R can be used for Retrieval Augmented Generation (RAG), a popular method to reduce the hallucination problem often encountered in LLM-generated content. RAG enhances text generation by providing the model with additional information that it can reference, improving the accuracy and grounding of its responses. In the case of Command R, this information is supplied in the form of "documents"[11]. Our initial plan was to compare the summarization results produced by Command R in both its simple and RAG modes and then choose the approach that provided the best quality.

According to Cohere, for optimal generation quality, the total word count of these documents should not exceed 300 words [12]. Therefore, we had to chunk the articles' text to provide news as documents to the Command R model. We opted for sentence-level chunking to preserve meaning. If chunking were done at the word level, sentences might be cut off mid-way, which could lead to a loss of context for the model. To achieve sentence-level tokenization, we initially considered using the **tokenize**[13] package from the **nltk** Python library. However, this library only supports three of the four languages we were working with, leaving out Ukrainian. For Ukrainian sentence tokenization, we instead used the **Tokenize UK**[14] Python library. We aimed to preserve the original paragraph structure as much as possible. However, when paragraphs exceeded the length recommended by Cohere, we balanced their size by re-splitting sentences into chunks of approximately ten sentences per chunk.

After generating and analyzing both versions of the summarizations, we found no clear distinction in overall quality between them. The most common type of hallucination we encountered involved the introduction of languages other than Ukrainian into the summaries, for example, Arabic, while still retaining the correct meaning. Unfortunately, DeepL API does not support partial translations (translating texts that are mainly in Ukrainian but contain patches in other languages), and using back-translation negatively impacts the accuracy of the summarizations. For example, "WHO" was mapped to "World Health Organization", which is correct in general but not in our case.

Given these issues, we decided to keep both options of summarization, leave language hallucinations as they were and instead implement a feedback mechanism in the interface. This feature allowed annotators to report if both provided summarizations made the annotation task impossible due to these errors. With this solution, we were ready to design and launch the crowdsourcing project.

[11] https://docs.cohere.com/docs/retrieval-augmented-generation-rag
[12] https://docs.cohere.com/reference/chat
[13] https://www.nltk.org/api/nltk.tokenize.html
[14] https://tokenize-uk.readthedocs.io/en/latest/readme.html

3.2. Crowdsourcing Project Design

There are several popular crowdsourcing platforms, such as Amazon Mechanical Turk[15], Appen[16], Clickworker[17], and others. We chose **Toloka** for several following reasons. First, Toloka was initially established as part of Yandex, a prominent Russian search company, over ten years ago. Consequently, the platform has attracted a large pool of annotators, many of whom speak languages from the Slavic family, including Ukrainian. Second, Toloka supports research initiatives and offers a grant program[18], which we applied to and were provided 500 dollars for our low-resourced NLP project. Toloka has a self-service platform[19] that allows users to independently configure labelling projects for various data modalities, including text, image, audio, and video. The platform provides various instruments for filtering annotators, designing quality control mechanisms, and creating dynamic project interfaces without requiring knowledge of CSS, HTML, or JavaScript, making it accessible for our project needs.

3.2.1. Glossary

Before presenting an overview of the labelling pipeline we designed on Toloka's self-service platform, we first provide a glossary of terms[20] specific to labelling projects on Toloka. Understanding these terms is crucial for correctly interpreting our pipeline.

- **Toloker**: An annotator on the platform.
- **Project**: A specific data labelling goal, represented by a single interface with a dedicated instruction. A project can include multiple pools.
- **Pool**: A data set to be labelled within a project, united by filters applied to annotators, per item pricing, and a set of quality control rules. A pool typically contains several tasks.
- **Task**: An individual labeling item. In our case, it refers to a pair of news articles along with all necessary information for annotation, as well as the corresponding annotation task. One task can be assigned to one or multiple annotators.
- **Assignment**: A labelling task that an annotator has accepted.
- **Overlap**: The number of Tolokers assigned to the same task to ensure multiple annotations for robustness. Overlap value refers to the number of annotators who should complete each task in the pool.

[15]https://www.mturk.com/
[16]https://www.appen.com/
[17]https://www.clickworker.com/
[18]https://toloka.ai/forms/grants/
[19]https://toloka.ai/data-labeling-platform/
[20]https://toloka.ai/docs/guide/overview/

- **Majority Vote**: A quality control rule that deems a response correct if the majority of Tolokers selects it. Other responses are marked as incorrect.
- **Manual Acceptance**: A pool setting that enables manual review of assignments. Rejected assignments within a predefined review period are not paid. Tolokers can appeal rejected assignments; if they are deemed correct upon review, assignments are accepted and paid.

3.2.2. Overview

Solutions on the Toloka self-service labelling platform can be created and managed through the API or in the platform's interface. Given the grant amount and the estimated number of news article pairs to label, working directly through the interface was more practical, especially since this was a one-time annotation project. Toloka's interface accepts data in **xlsx**, **json**, and **tsv** formats, and it was straightforward to convert our scraped data to fit them. We decided to divide a labeling project into different pools based on language combinations, which allowed us to train and assess annotators separately based on their language skills. However, language selection did not impact the pricing or quality control methods.

We designed a Tolokers funnell using the following steps. First, every Toloker interested in the project and meeting the language requirements was required to complete a non-paid training. This training allowed them to study the instructions and review examples of how tasks should be labelled, with explanations for each labelling decision provided. The training interface differed from the main task interface, so we set it up as a separate project. This difference is explained in the "Interface" Section. Second, after reviewing the training examples, each Toloker had to pass an exam identical to the main tasks to demonstrate their understanding of the instructions. Since the exam was the same as the main tasks in terms of interface and project setup, we paid Tolokers for the exam. Upon passing the exam, they gained access to the main assignments.

Since a part of the task involved text generation (e.g., explaining the level of similarity between events in news articles), it was challenging to design fully automated quality control. However, our goal was to automate the process as much as possible to ensure scalability. For this reason, we integrated Large Language Models (LLMs) as a support tool and added an overlap of **three** annotators per news article pair. The overlap allowed us to use majority voting wherever possible. The "Quality Control" section further discusses our quality control strategy.

We aimed to create a fair-paying project with an intuitive interface, detailed instructions, and a clear feedback loop, allowing annotators to appeal rejected tasks.

3.2.3. Instruction

In this section, we provide the Instructions we used for the training and the main project. We provide them translated to English, as the original language of instructions is Ukrainian.

Training

`Instruction for Training, translated to English`

This is the pre-project training. In it, we show you examples of completed tasks and corresponding explanations. Your goal is to familiarize yourself with the instructions and then study them deeper with the help of examples. After completing the training, you will gain access to the main project, where the assignments are paid. You will be provided a paid exam, which, when passed, will give you access to all the main tasks. Thank you for your participation! Your input helps us to improve the quality of the news.

Disclaimer *We assume that if you have stated certain languages (Polish, Ukrainian, Russian or English) in your profile, you are able to complete the tasks in them.*

Task description

In each task, you will compare two news articles and evaluate their similarity.
The news articles in the pair may be in different languages, but this should not affect your comparison. For example, Wołodymyr Zełenski (Polish) and Volodymyr Zelenskyy (English) are the same actors in an event.
For each news article, we have provided:

- A link to the news website (be sure to follow it to see the original article, at least the headline and date of publication);
- Full text of the article (if you want to check some details right in the task interface);
- Two summarization options made by specially trained algorithms (including basic information on four main parameters of events: WHO, WHEN, WHERE and WHAT happened)

Assessing news similarity

You will answer four questions to help us draw a conclusion about the similarity of the news. Examples of answers to these questions with explanations will be shown throughout this training.

Do the main characters in the news match? (WHO)
The actor of an article can be a specific person (Volodymyr Zelenskyi), an organization/party or any other association of people.

- **Exactly** - the main and secondary actors in the articles are the same. At the same time, it is important to note that one actor can be called differently, for example, "Volodymyr Zelenskyi" and "President of Ukraine".
- **Partially** (there are overlaps) - Some of the main or secondary actors are the same.

- **No** - The protagonists in the articles do not coincide at all; they have nothing in common.
- **At least one article does not have any actors** - for example, one of the articles states something very vague, e.g. "the entire population of the Earth", or the article is a weather forecast.

Do the places where the events in the news occur match? (WHERE)

- **Exactly** - they are talking about the same place. If one article mentions a specific place/city/country, it should be mentioned in the second article. If one article mentions only a country and the other a city, this is not an exact match. In this case, the same place may be referred to differently, such as "St. Petersburg" and "St. Petersburg".
- **Partially** (there are overlaps) - The place mentioned in one article is located within the boundaries of the place mentioned in another article. For example, if one article refers to events taking place in Ukraine and the other in Kyiv, this option is selected. Also, if different places (e.g., countries) are mentioned in the articles and some of the names of the places are the same.
- **No** - The places are entirely different (different places in the city, in the country, different countries).
- **At least one article does not mention the location at all** - something very vague, for example, something that happens worldwide, all over the Internet, or if the location is not specified. However, suppose the place of action can be understood, for example. In that case, the article says "in our country," and the author is obviously a Ukrainian, then, this option should not be selected. Do the times of the events in the news coincide?

Do the times of the events in the news coincide? (WHEN)

It is important to understand that we are interested in the time of the event in an article, NOT the publishing date. However, the article publication date can be very useful in determining the time of the event. Therefore, we ask you to follow the link to the original news article to use this information.

- **Same event in time** - it is obvious from the article that it is about the same event that happened simultaneously.
- **Within one month** - two events occurred within the same month.
- **More than a month apart** - the difference between the times of the events is more than a month.

- **From at least one article, it is impossible to establish the time of the event**. Even if the article's publication date is present, which is almost always the case. If, after reading the text of the article, it is impossible to determine when the event itself took place, or the time period in the article is very vague, for example, it states "at all times".

How do news topics/events relate to each other? (WHAT)

- **The same**
- **Common specific topic**
- **Different**

Same

News articles describe the same event in the same place, with the same people, at approximately the same time.
However, the events in the news may be described as follows:

- in a different tone/style;
- using different words/slang;
- with a different focus (one focuses more on one part of the event, the other on another);
- from different points of view, political position;
- with errors (to your knowledge) in the details, which makes the news seem different

If you are not 100% sure that the news is about the same event with the same actors, place and approximate time, then you should choose the option "Common specific topic".

Common specific topic

News stories are related to one narrow topic or one joint event but describe different incidents within it.

Examples:

- Curfews imposed due to COVID-19 (in different countries or changes in its status within the same country);
- Robbery of a grocery store done by teenagers (in different places or different events within the same major case);
- Joe Biden's 2020 election campaign and events within it.

News topics should be narrow and specific, so, for example, two news stories on broad and popular news area topics such as COVID-19 (in general), death (in general), and elections (in general ;) do not have to have any commonality in themselves.

Different
Events described in the news articles have little to do with each other. They either do not have a common theme, or the common theme is broad (e.g., COVID-19), and it is not a reason to consider the news relatable.

Justify your choice

Examples of choice justification are shown in the training.
If news articles:

- **are the same:** briefly describe, summarizing in 1-2 sentences, what event both articles mention.
- **united by a specific topic:** briefly describe this topic in 1-2 sentences.
- **are different:** briefly summarize in 1-2 sentences why, from your point of view, these topics are different (one is about ..., another is about ...)

The instructions for the main project are mostly identical to the ones for the training, with the only following difference:

Exam & Main pools

`Instruction for Exam & Main pools, translated to English`

Same as instruction in Training

Payment

We check and accept all assignments within the span of **10 days**. Incorrect assignments will be rejected. We take the subjectivity of grades into account. You can file an appeal if you believe your assignment has been rejected unfairly.

Task description

Same as instruction in Training

Assessing news similarity

Problems with news articles' evaluation
If you encounter such problems as:

- Instead of summarization, there is some kind of error;

- It seems to you that it's not a news article in front of you but something else;
- An unfamiliar language (not the one you completed the Training on) prevails in the text of the news article/summaries so that it is impossible to understand what is being said;
- The link to the news article does not open (you waited a minute+);
- Any other problem.

The interface has a button "*There's a problem with evaluating this news article*" below each news article.
When you click on it, you can choose one of the existing options or select the option "*Other*" and describe a problem that is not listed.
In this case, you do NOT need to label the news articles; the task will be fully paid if we reproduce the error you specified when checking the task.

Same as instruction in Training

3.2.4. Interface Design

A well-designed interface is essential for high-quality labelling. An intuitive interface not only reduces labelling time but also helps annotators to stay focused. Moreover, the annotation quality is further increased when instructions are subtly embedded within the interface. A poorly designed or buggy interface can slow down the process and negatively impact the quality of the labels.

Toloka offers a Template Builder[21] tool for creating dynamic interfaces using intuitively formed components in JSON format, eliminating the need to learn HTML or JavaScript for interface design. Template Builder supports auto-completion with helpful hints and preset code snippets, making the design process more manageable.

We designed two projects, each with its own interface for training and main tasks. While the interfaces differed in the part intended for Tolokers input, they shared a common component for representing the news article pairs. This component includes the placement of the news texts, links to the original news sources, and two versions of the summarizations per article. The English translation of this part of the interfaces is shown in Figure 3.2.

1. **Project Name**: "News comparison (Training)"
2. **Disclaimer**: "In the process of labelling, you may come across sensitive or disturbing topics, as news articles are rarely neutral. If you feel that you are not ready to work with such materials, please refrain from participating in this project."
3. **Reminder**: "Read the instruction carefully before completing the task"

[21] https://toloka.ai/docs/template-builder/

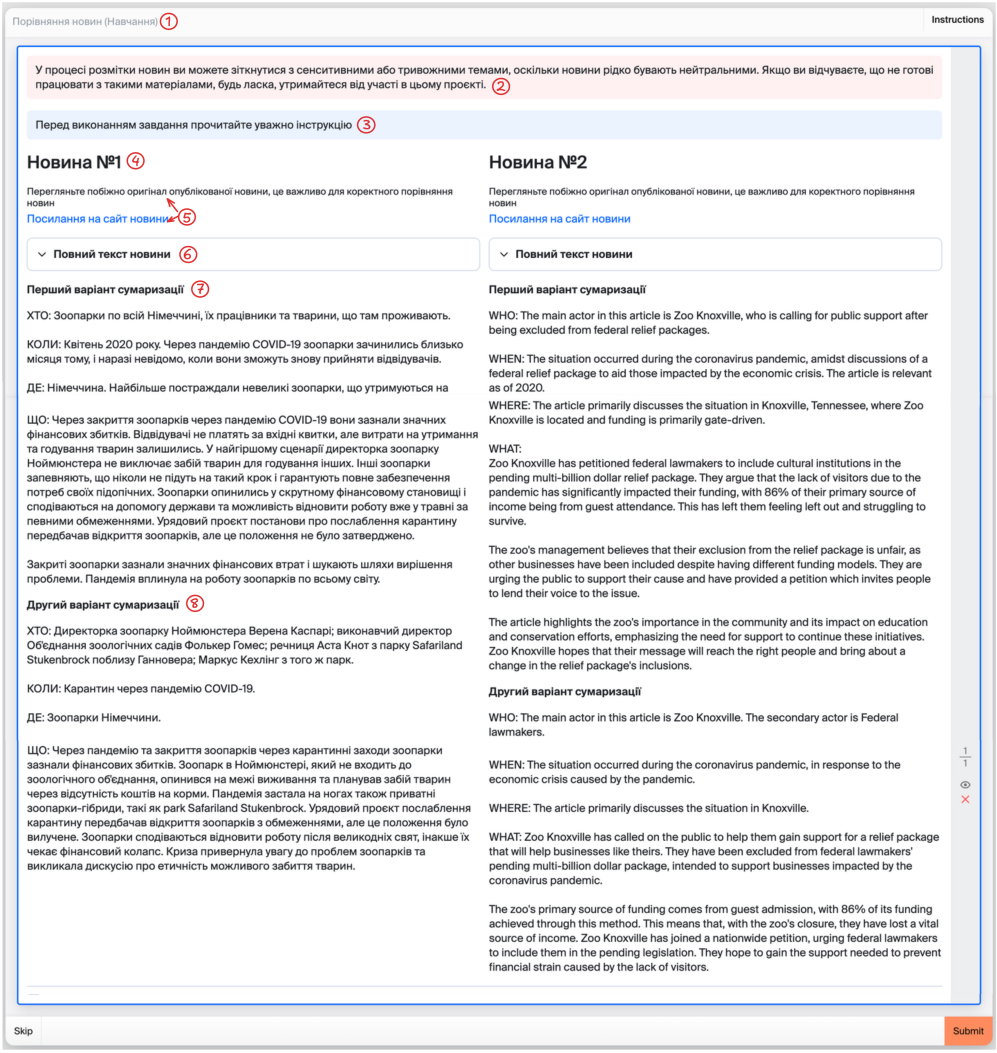


Figure 3.2.: Interface, comparison of news articles, Shared Part

4. **News article header & Reminder** News Article №1. Skim through the original news article briefly; this is important for correctly comparing news articles.
5. **Link to the news article**
6. **Full text of the news article**
7. **The first version of summarization**
8. **The second version of summarization**

Training

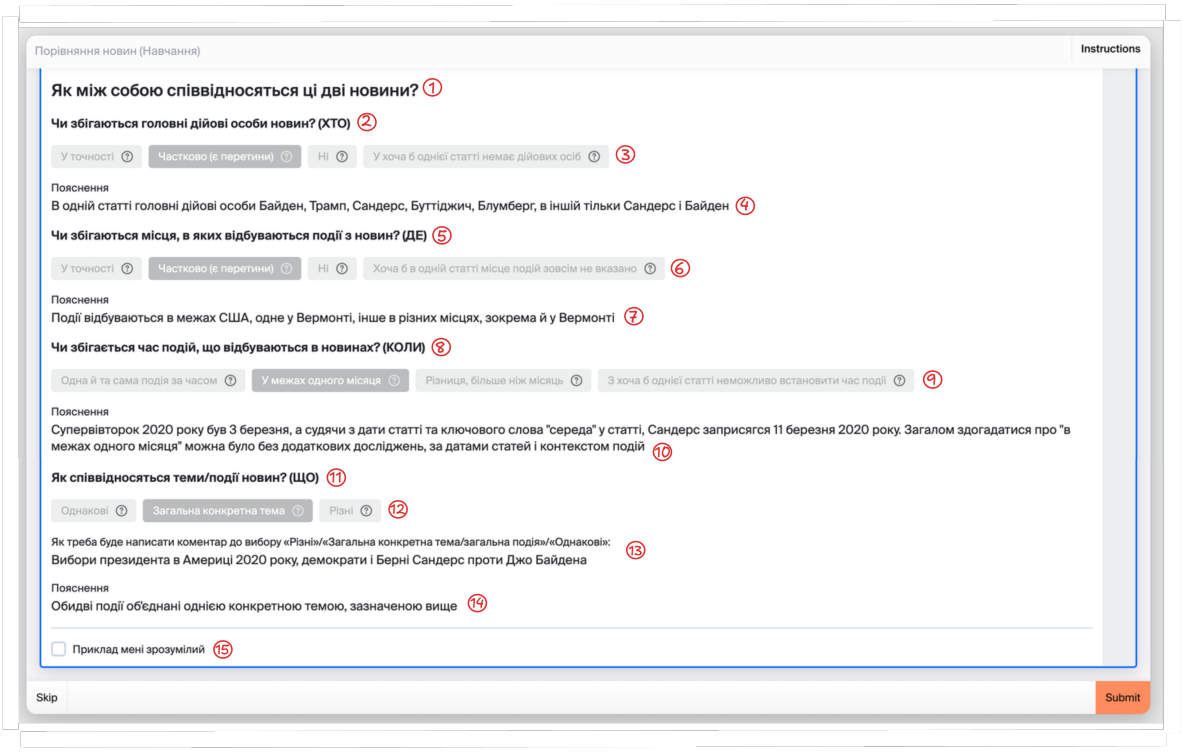


Figure 3.3.: Interface, comparison of news articles, Training

The training project differs from the main project in part in that it provides Tolokers with labelling examples and explanations of every choice they are expected to make in the main project. In the training sections, the only action required is to select the checkbox "I understood the example." If something is unclear, Tolokers have the option to contact the project designers through messages on the Toloka platform. The translation of this part of the training interface, shown in Figure 3.3, is provided below.

1. **How do these two news articles relate to each other?**

2. **Do the main actors in the news articles match? (WHO)**
3. **Options (from left to right)**: Exactly; Partly (there are overlaps); No; At least one article has no explicitly defined actors. *Question marks are action buttons showing hints, explaining the meaning of each option*
4. **Explanation**: In one article, the main actors are Biden, Trump, Sanders, Buttigieg, and Bloomberg. In the other, only Sanders and Biden are mentioned.
5. **Do the places where the events in the news are taking place match? (WHERE)**
6. **Options (from left to right)**: Exactly; Partly (there are overlaps); No; At least one article does not mention the place of the events.
7. **Explanation**: The events take place within the United States, one in Vermont, the other in various locations, including Vermont
8. **Do time ranges of events in the news coincide with each other?**
9. **Options (from left to right)**: The same event in time; Within one month; More than a month difference; It is impossible to determine the time of the event from at least one article.
10. **Explanation**: Super Tuesday 2020 was on March 3rd, and judging by the date of the article and the keyword “Wednesday” in the article, Sanders vowed to continue his campaign on March 11, 2020. It was generally possible to guess “within one month” without additional research based on the articles’ dates and the events’ context.
11. **How do news topics/events relate to each other?**
12. **Options (from left to right)**: Similar; Common specific topic; Different;
13. **Labeling example**: How to write a comment to the "Similar"/"Common specific topic/event"/"Different" choice: "American presidential election 2020, Democrats and Bernie Sanders vs. Joe Biden"
14. **Explanation** Both events are united by one specific topic mentioned above
15. **The example is clear to me**

Exam & Main pools

In main pools, in case there are problems with news articles that stop labellers from performing their task, we introduced, under each news article in the pair, the dynamic expandable part of an interface, presented in a fully expanded form in Figure 3.4. Here’s the translation of it:

1. **There’s a problem with evaluating this news article** *Question mark is an action button explaining what’s meant by the "problem"*

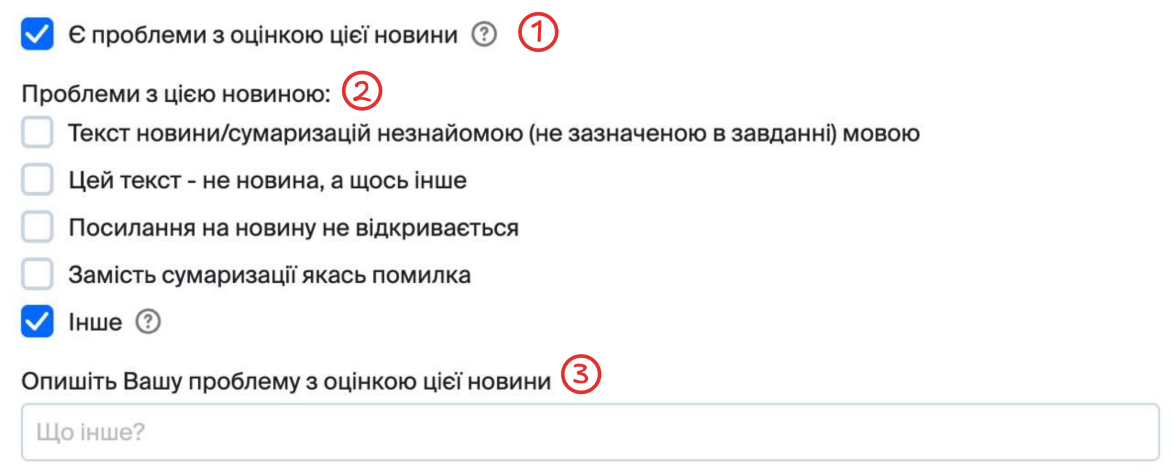


Figure 3.4.: Interface, comparison of news articles, Errors Detection

2. **List of problems**: Problems with this news article:
   - Text of news article/summaries is in an unfamiliar language (not specified in the assignment)
   - This text is not a news article but something else
   - The link to the news article does not open
   - Instead of summarization, there is some kind of error
   - Other *Question mark is an action button explaining when to select "Other"*
3. If "Other" was chosen: Describe your problem with the evaluation of this news article: "What's other?"

Suppose the labeller chose that there's any problem with at least one news article in a pair. In that case, the interface allows to skip the selection of any answers to the main questions presented in Figure 3.5. If the news article actually had a problem defined by the labeller, the task was fully paid for.

Another key difference between the main and Training interfaces is the dynamic, expandable labelling section (Figure 3.5). This section provides hints to remind Tolokers of the instruction requirements, but unlike the training interface, it is left unfilled and is expected to be completed by the Tolokers. They are required to provide comments explaining their choices based on their answer to the WHAT question. The translation of this part of the main interface is provided below.

1. What are both news articles about?
2. What is the common specific topic of the news articles?
3. Why are the news articles different?

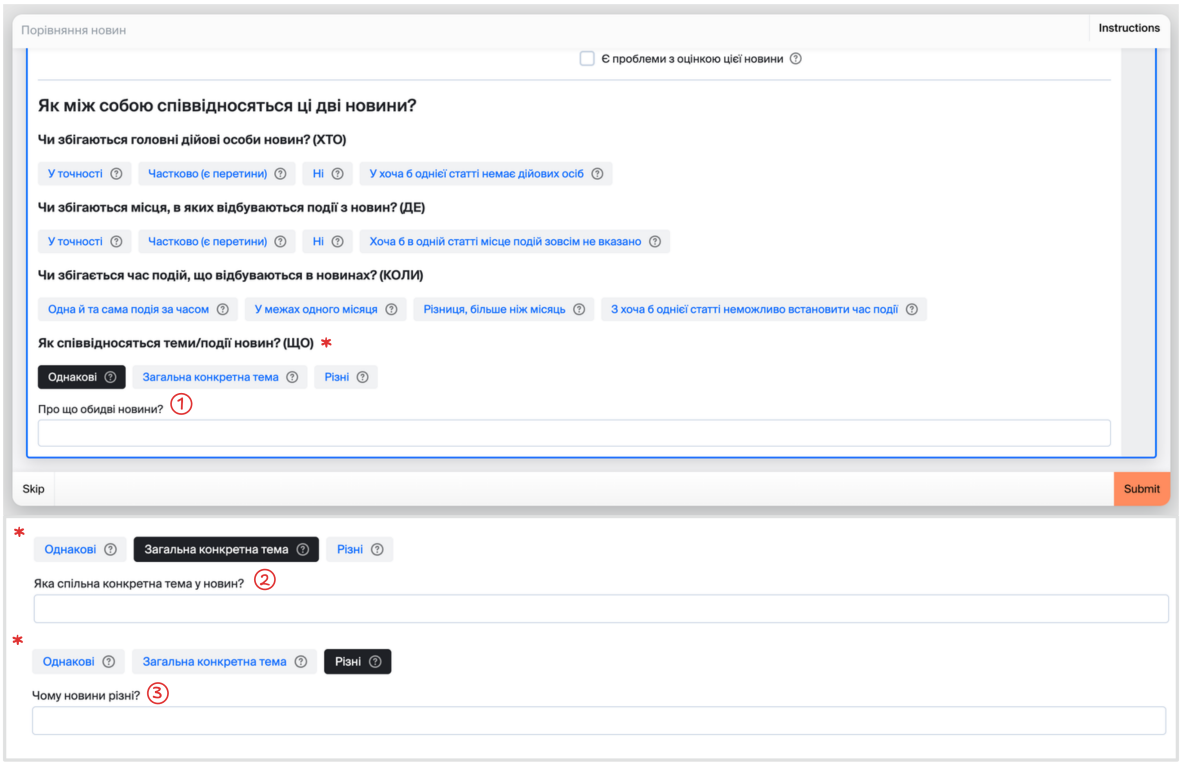


Figure 3.5.: Interface, comparison of news articles, Exam & Main pools

Interfaces are designed so that annotation can't be submitted unless completed fully, as well as if original news sources of news articles weren't checked.

3.2.5. Quality Control

Quality control in Toloka is a system for monitoring and managing task performance to improve response accuracy and restrict access for bad performers.

We built our quality control process on top of Toloka's system, adding our mechanisms and dividing it into two stages: **before labelling**, which ensures that we filter out annotators who have not studied the instructions or labelling interface thoroughly, and those who have not demonstrated an understanding of the labelling task or the necessary language skills; and **after labelling**, where we filter out assignments that do not follow our instructions and aggregate results using majority voting, which helps manage the subjective nature of the labelling tasks.

During labelling, the only quality control rule we applied was the *"Fast responses"* rule for the training phase to filter out labellers who were not carefully reviewing the articles. This rule restricted access for Tolokers, who responded too quickly. If any three training tasks in a pool were completed in under 60 seconds, we banned the Toloker from the project.

Before labelling

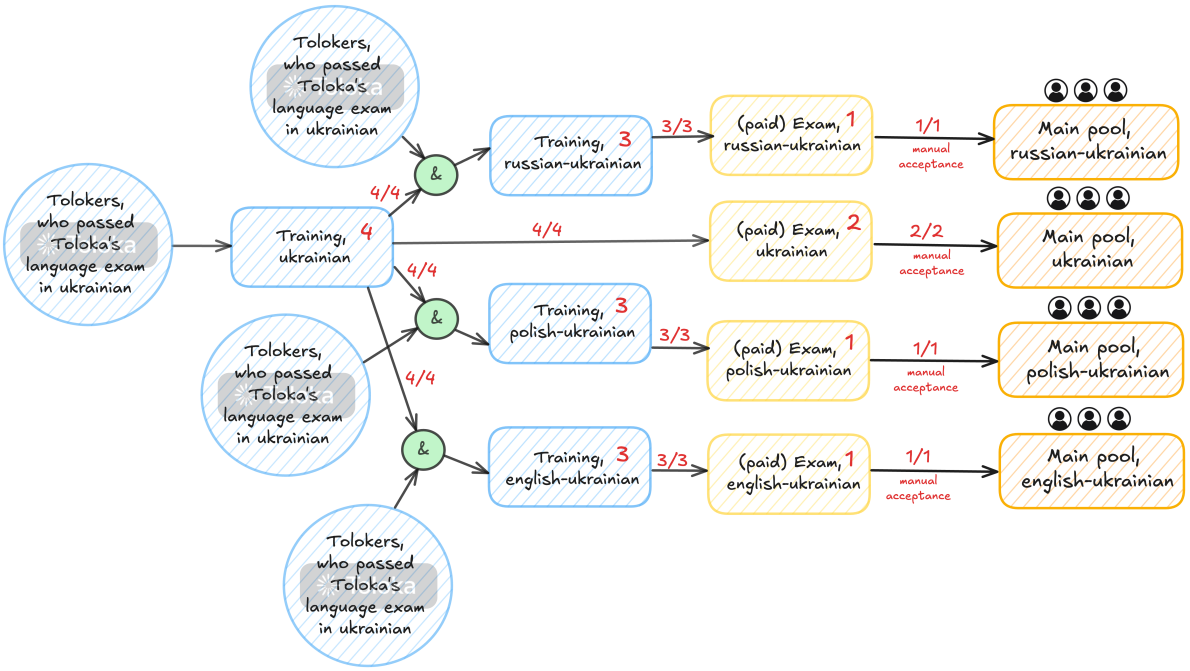


Figure 3.6.: Annotators Selection

Quality Control pipeline part before labelling is sketched in Figure 3.6.

Toloka offers mechanisms to pre-filter Tolokers who have passed an official platform test in specific languages. We utilized these filters to select individuals proficient in English, Russian, Ukrainian, and Polish.

Based on these filters, Tolokers were granted access to training pools in the respective languages. However, before they could proceed with training in other languages, each Toloker was required to complete the training in Ukrainian since Ukrainian is the primary language for all pools, as well as for the interface and instructions. During the Ukrainian training, we provided four news article pairs, representing a variety of possible answer options: one "Similar," two "Somewhat Related," and one "Different". After successfully completing the training in Ukrainian, where the Tolokers had to demonstrate comprehension of the examples, they could proceed with similar training in other languages (three assignments per language) or directly take a paid exam in Ukrainian. After completing each training session, Toloker unlocked access to the corresponding paid exam in the respective language.

The Ukrainian exam consisted of two news article pairs: one labelled as "Similar" and the other as "Different". In contrast, the Russian, English, and Polish exams consisted of only one assignment. The limited number of assignments in each exam was chosen due to the manual exam review process, which took into account the subjective nature of the tasks. This setting made it difficult for Tolokers to "cheat," especially since the assignments required generating text based on the given instructions. If all assignments in the exam were accepted, the Toloker would gain access to the corresponding main pool and could start working on paid labelling tasks.

After Labelling

Quality Control pipeline part after labelling is sketched in Figure 3.7.

In Toloka, it is possible to manually review labelled tasks directly on the platform. Reviewers are provided with an interface similar to the labelling project interface, where they can see responses from individual labellers and decide whether to "Accept" or "Reject" them. However, we quickly realized that manually reviewing five hundred pairs of news articles was impractical, especially when labelled with an overlap of three, resulting in 1500 pairs. This situation demanded a scalable solution.

However, there were no automatic quality control rules in Toloka that could directly assess the text generated by labellers. The task is quite complex since the number of possible correct answers in text generation is vast, if not infinite. We could apply the Majority Vote quality control rules for the classification tasks, where answers to the WHO, WHEN, WHERE, and WHAT questions are limited to four, four, four, and three options, respectively. However, justifying the answer to the WHAT question through comments could not be verified using this method.

To address this, we combined the advantages of the Majority Vote quality control model with the semantic reasoning ability of large language models (LLMs), using manual review only in a limited number of situations. These situations included cases where all three labellers disagreed, the LLM rejected justifications provided by labellers, and Tolokers appealed rejected assignments. The Toloka interface provides the option to upload the results of

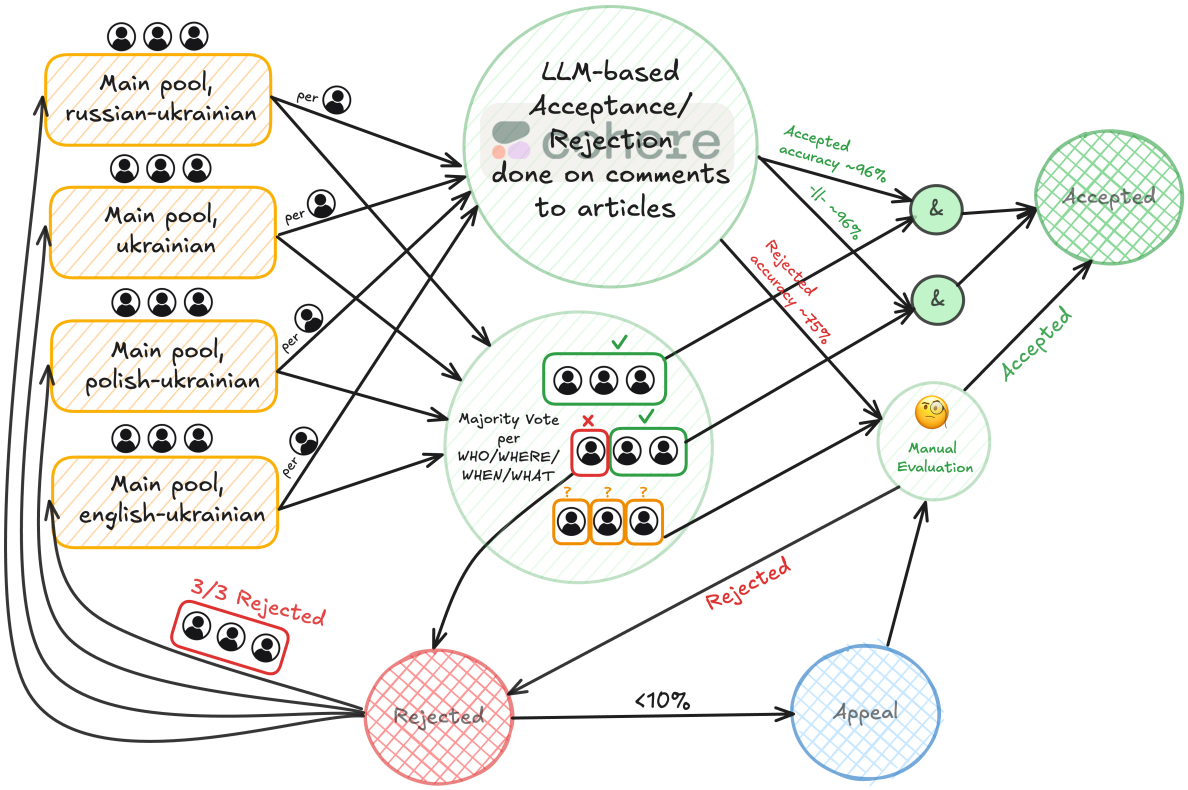


Figure 3.7.: Quality control of labelled assignments overview

manual reviews via a file. Instead of manually inputting results through the interface, we took advantage of this feature.

For the LLM model, we again used **Command R** by Cohere. We analyzed a sample of the comments to identify common mistakes, particularly those resulting from labellers ignoring guidelines on providing justification comments. We categorized these mistakes into two groups based on the similarity of events in the news article pairs: "Different" and "Somewhat Related". For "Similar" news pairs, we automatically checked whether the answers to the WHO, WHERE and WHEN questions were all "yes." If not, we manually reviewed these cases since the number of news pairs labelled as "Similar" was manageable. However, for future projects, we recommend using LLMs to check the comments for "Similar" news pairs as well since these can also contain mistakes related to overlooking instruction guidelines.

The prompts used for checking the comments in these two groups are as follows:

Prompt, Acceptance-Rejection with Command R Cohere, "Different"

```
## Instruction
    You are provided with a comment explaining
    why the news articles in a pair are different.
    Your goal is to judge if, out of this comment, one can understand
    what these two news articles are about (Yes), or if
    the comment just says that the news articles are different (No).

    ## Examples
    For instance, out of the following comments:
    - "A lawsuit from journalists and the trial of a police officer"
    Yes, both news are briefly summarized.
    - "Martial arts in the U.S.A.;
    a review of five old martial arts fighters"
    Yes, both news are briefly summarized.
    - "The first article reports that Dagestan has launched
    the production of oxygen valves for medical equipment
    due to the increased demand caused by the pandemic.
    The second article is about a ‘‘counterterrorist operation’’
    in Makhachkala and Kaspiysk."
    Yes, both news are briefly summarized.
    - "First covid-19 in Ukraine, second - parade in Russia."
    Yes, both news are briefly summarized.
    - "Assaulting a pensioner and the weather"
    Yes, both news are briefly summarized.
    - "Events are different"
    No, it just says that news articles are different.
    - "Completely different events"
```

```
    No, it just says that news articles are different.
    - "The events are not related in any way"
    No

    ## Input
    - "{COMMENT}"
```

Prompt, Acceptance-Rejection with Command R Cohere, "Somewhat related"

```
## Instruction
    You are provided with a comment written by a labeller
    explaining why news articles in a pair of news articles
    are both on one specific topic.
    However, some labellers did not read the instructions,
    which says that COVID-19/elections/death, in general,
    is too broad of a topic to be common for news.
    If news articles both share a specific subtopic of COVID-19
    e.g. death statistics, recent anti-COVID measures, the new vaccine,
    new stamm; a specific subtopic of elections,
    e.g. elections in the USA 2024, elections in Russia 2024;
    a specific subtopic of death, e.g. death in war or
    death from murder on a street,
    only then articles are both on a specific topic.
    Your goal is to judge if labellers did read the instructions (Yes)
    or did not (No).
    Also, if they did not read the instructions,
    they won't specify the specific topic.
    They will just write something like "common (specific) topic".

    ## Examples
    - "Both articles are on the topic of the coronavirus pandemic"
    No, the labeller did not read the instructions;
    it's just a comment about COVID-19.
    - "The common topic of covid and post covid"
    No, the labeller did not read the instructions;
    it's just a comment about COVID-19.
    - "Pandemic"
    No, the labeller did not read the instructions;
    it's just a comment about COVID-19.
    - "common theme"
    No, it does not say anything about the topic.
    - "A new strain of coronavirus"
```

```
Yes, it's not only about COVID-19 but about the new strain of it
- "Common topic elections"
No, the labeller did not read the instructions;
it's just a comment about elections
- "Research on COVID-19"
Yes, it's not only about COVID-19 but about research made on it
- "Both topics are about panic over covid"
Yes, it's not only about COVID-19
but also about the panic around it
- "Emergency landing of an aeroplane"
Yes, it's a specific topic, not a generic one
like COVID-19/elections/death
## Input
- "{COMMENT}"
```

We assessed the quality of Cohere's judgments on a subsample of the data. The rate of False Positives was **96%**, while False Negatives were approximately **75%**. These percentages indicate that using Large Language Models (LLMs) as a moderation tool together with human oversight holds significant potential for improving quality in crowdsourcing projects.

Additionally, we analyzed the rate of appealed rejections, which was less than **10%**. This low percentage suggests the possibility of scaling the project in the future.

Tasks that received three out of three rejected labels were sent back *once* for relabeling. To ensure the highest labelling quality for news pairs with three out of three rejections, we selected annotators who had shown the most engagement in the project and had a high percentage of accepted assignments. These annotators were identified using the *"Results of a manual review"* rule[22] in Toloka. Out of five hundred pairs of news articles, nine were Polish-Ukrainian, sixteen were English-Ukrainian, eight were Russian-Ukrainian, and twenty-seven were Ukrainian-Ukrainian, all requiring relabeling due to disagreement. For these pairs, we only kept the last three out of the six labels provided.

Overall, using automated quality control combined with human oversight, it is feasible to scale crowdsourcing projects even for complex tasks like ours while maintaining high-quality results.

3.3. Results

The results of the labelling process have been published as an open-source dataset on HuggingFace[23]. Before publication, we anonymized sensitive data, such as the labellers' IDs on the platform, and aggregated the information. A detailed description of the final dataset is provided in the following sections.

[22]https://toloka.ai/docs/guide/reviewing-assignments/
[23]https://huggingface.co/datasets/tum-nlp/multilingual_news_SS_uk_ru_en_pl

The code for scraping and preparing the data, as well as labelling projects' design, quality control scripts, dataset final aggregation and statistics calculation, are available in a GitHub repository[24]. We encourage researchers to contribute and enhance the code for further scaling of the news articles semantic similarity labelling tasks.

3.3.1. Labeling Characteristics

Toloka provides annotators with a quick instrument of feedback to express their opinion on the design and configuration of the labelling project, if it's fair pay-wise and if the organizers of the task are responsive and helpful. Our projects received high overall ratings[25]: **4.98/5.00** for the Main Project and **4.95/5.00** for the Training project.

Category of Spending	**Per Assignment ($)**	**Total ($)**
Exam assignments	0.20	24.08
Main pools, accepted assignments	0.20	250.04
Bonus, rejected assignments with comments relevant to instructions	0.15	129.95
Additional bonus, accepted assignments	0.06	79.40
Total		**464.16**

Table 3.1.: Total budget of labelling

We balanced the assignment payment, considering the grant funding and the minimum hourly wage in Ukraine, which is currently **1.12** USD. For accepted assignments in the exams, annotators received **0.2** USD, and in the main pools, **0.26** USD. For assignments that were rejected due to the Majority Vote Quality Control Rule but where annotators provided comments based on the instructions and training recommendations, we paid **0.15** USD. The overall budget of the project is presented in Table 3.1.

The funnel of labellers is shown in Figure 3.8. It is important to note that we limited the number of labellers in our projects, stopping the training projects once we had enough participants based on our estimates. This was due to the limited amount of grant money available, so the numbers presented do not fully reflect the potential scalability of the project.

Out of the **43** Tolokers who submitted work in the main project (including exams and main pools), **29** participated in the main pools, with an average of **7** active users per day. The median age of the labellers was **49** years, ranging from a minimum of **28** to a maximum of **74**. The highest number of assignments submitted by a single person was **138**, while the median number of submissions per person was **52**. On average, completing a labelling task took **9.06** minutes. These values were calculated based on the data shown in Figure A.6 and Figure A.7, which were generated through Toloka's project labelling interface[26].

[24]https://github.com/mrscoopers/tum-nlp-multilingual_news_SS_uk_ru_en_pl
[25]https://toloka.ai/docs/guide/project_rating_stat/
[26]https://toloka.ai/docs/guide/project-statistic/

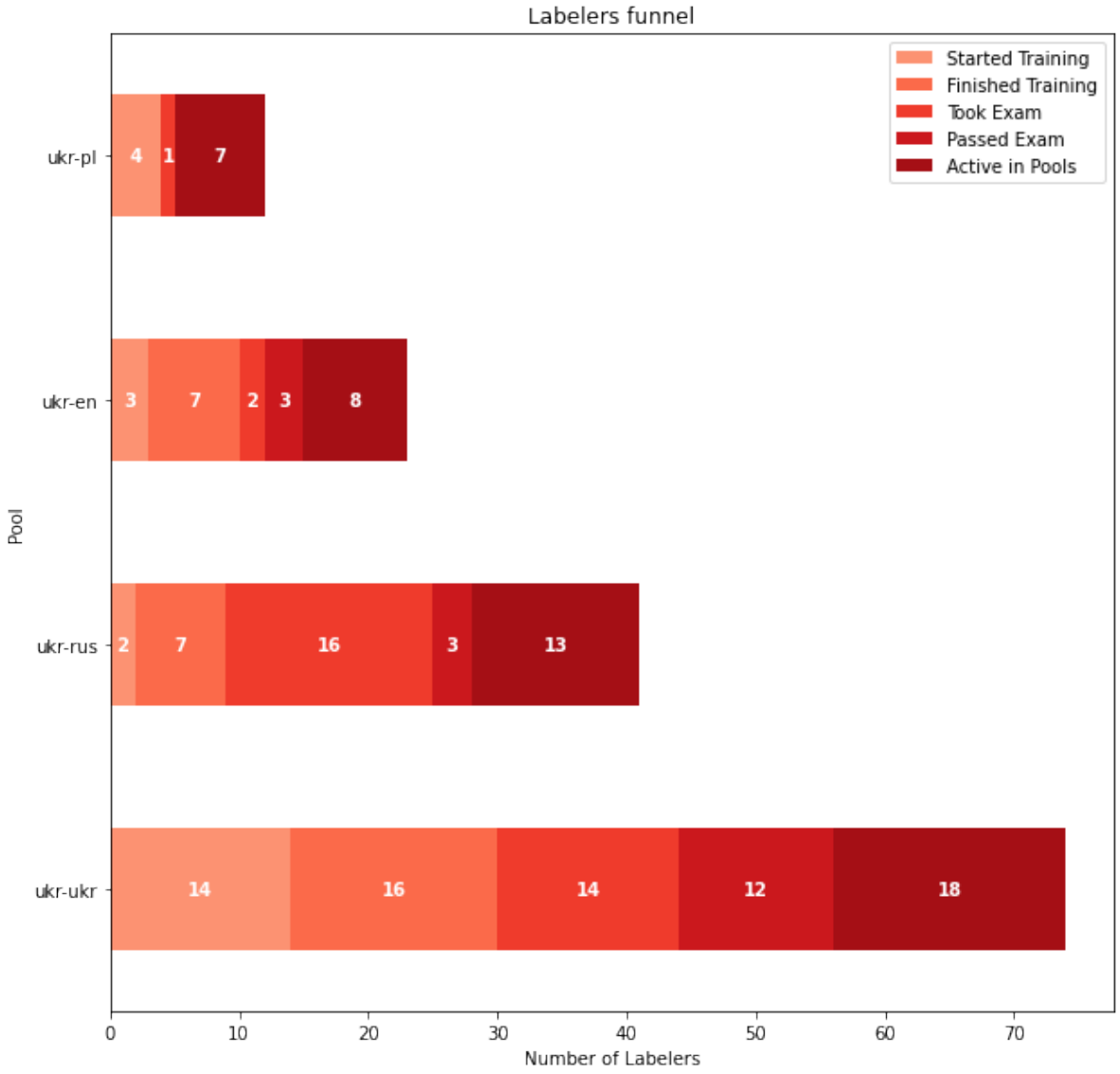


Figure 3.8.: Funnel of labellers per pool

3.3.2. Final Dataset Description

Here's the description of the final open-sourced dataset:

- **label_1**
  - **labeller** Labeller (anonymized) №X $\in [1, 30]$
  - **who_same** Labeller's answer to the question "Do the main characters in the news match?" mapped to values "yes"/"partly"/"no"/"incomparable."
  - **when_same** Labeller's answer to the question "Do the places where the events in the news occur match?" mapped to values "yes"/"partly"/"no"/"incomparable."
  - **where_same** Labeller's answer to the question "Do the times of the events in the news coincide?" mapped to values "yes"/"partly"/"no"/"incomparable."
  - **what_same** Labeller's answer to the question "How do news topics/events relate to each other?" mapped to values "similar"/"somewhat_related"/"different."
  - **what_comment** Labeller's justification of the answer to the question "How do news topics/events relate to each other?"
  - **what_comment_translated** Since in the project there were no limitations on the language used for the comments, labellers used languages besides Ukrainian for answer justification. For consistency, we translated all comments to Ukrainian using DeepL.
  - **comment_is_good** If the comment was evaluated as good by the LLM-based script described in Section 3.2.5, or an expert when the labelling was not processed automatically. *Values*: True/False.
  - **processed_automatically** If the labelling by this annotator was processed automatically by the Majority Vote rule and LLM script described in Section 3.2.5, or it was evaluated manually by an expert. *Values*: True/False
  - **verdict** If the labelling by this annotator is accepted or rejected. *Values*: "+"/"-"
  - **comment** If the labelling by this annotator was rejected, the reason for rejection; otherwise empty.
- **label_2** Analogously to label_1
- **label_3** Analogously to label_1
- **text1** Full text of the first news article in the pair, parsed by Newspaper3k library.
- **text2** Full text of the second news article in the pair, parsed by Newspaper3k library.
- **source1** Link to the source of the first news article in the pair.
- **source2** Link to the source of the second news article in the pair.
- **url_lang1** Language of the first news article in the pair. *Values*: "uk"/"en"/"pl"/"ru"

- **url_lang2** Language of the first news article in the pair. *Values*: "uk"/"en"/"pl"/"ru"
- **language_of_pair** Language of the pair. *Values*: "ukr-ukr"/"ukr-en"/"ukr-pl"/"ukr-ru"
- **summ_text1** First version of summarization of the first news article in the pair described in Section 3.1.2
- **summ_text2** First version of summarization of the second news article in the pair described in Section 3.1.2
- **summ_with_documents_text1** Second version of summarization of the first news article in the pair described in the Section 3.1.2
- **summ_with_documents_text2** Second version of summarization of the second news article in the pair described in the Section 3.1.2
- **aggregated_who** Aggregated by majority vote method (Section 3.2.1) from three available annotations answer to the question "Do the main characters in the news match?" mapped to values "yes"/"partly"/"no"/"incomparable"/"disagreement"
- **aggregated_where** Aggregated by majority vote method (Section 3.2.1) from three available annotations answer to the question "Do the times of the events in the news coincide?" mapped to values "yes"/"partly"/"no"/"incomparable"/"disagreement"
- **aggregated_when** Aggregated by majority vote method (Section 3.2.1) from three available annotations answer to the question "Do the places where the events in the news occur match?" mapped to values "yes"/"partly"/"no"/"incomparable"/"disagreement"
- **aggregated_what** Aggregated by majority vote method (Section 3.2.1) from three available annotations answer to the question "How do news topics/events relate to each other?" mapped to values "yes"/"partly"/"no"/"incomparable"/"disagreement"

3.3.3. Dataset Statistics

The resulting dataset contains **500** pairs of news articles, with the language distribution shown in Figure 3.9. The level of agreement, calculated based on the overlap of three annotations, is illustrated in Figure 3.10. This high level of agreement demonstrates that, while the news semantic similarity task is non-trivial, it is well-suited for crowdsourcing labelling.

Although we aimed to balance the distribution of news article pairs with varying levels of similarity in the final dataset through our scraping pipeline design, the majority of the pairs contain different events with different actors, as shown in Figure 3.11. However, we still believe this dataset is a valuable contribution to the Ukrainian Natural Language Processing domain for two main reasons. First, the dataset includes textual justifications provided by Tolokers for their answer choices, offering a high level of explainability. These justifications could be used to fine-tune machine-learning models on various tasks within the news domain. Statistics based on these generated comments are shown in Figure 3.12 and Figure 3.13.

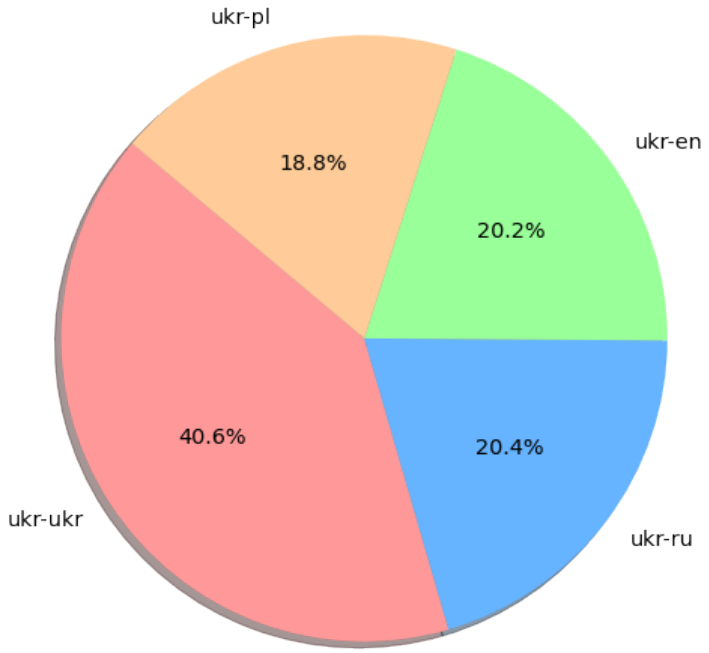


Figure 3.9.: Distribution of language pairs in the final dataset

Second, the entire pipeline could serve as a foundation for future large-scale projects focused on gathering news semantic similarity data. The pipeline is also adaptable to other languages, making it a valuable contribution to multilingual NLP.

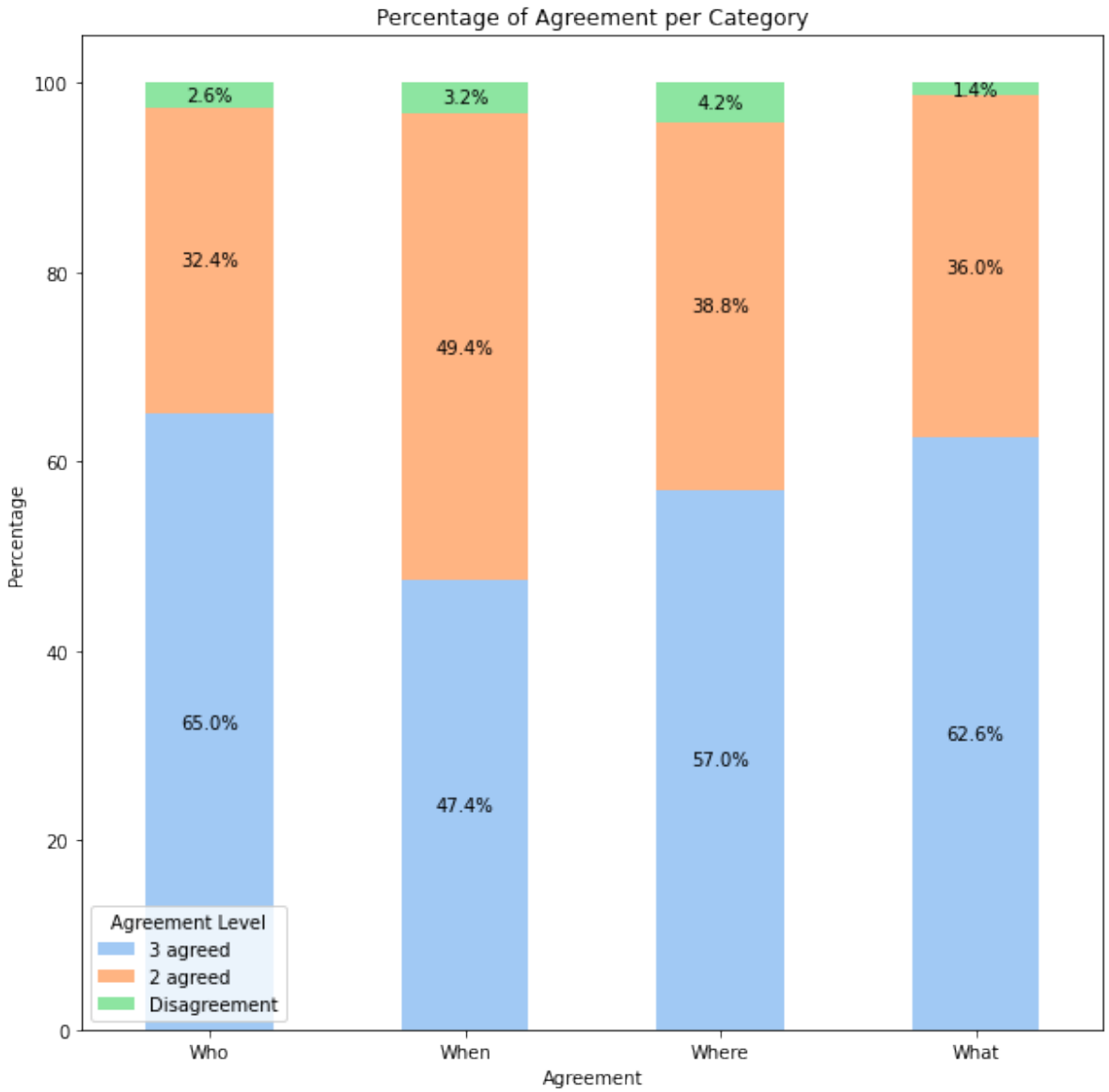


Figure 3.10.: Level of labelers agreement

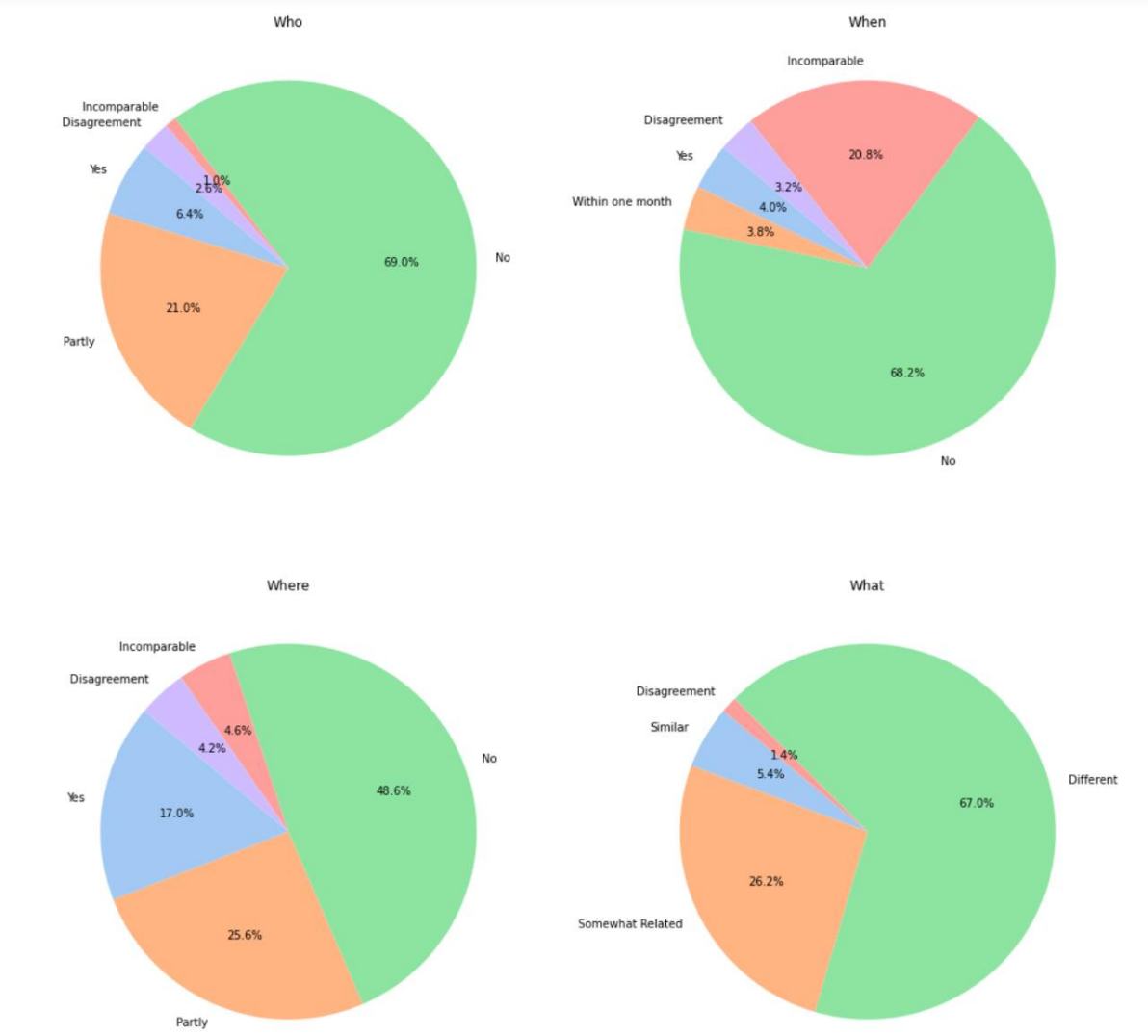


Figure 3.11.: Level of labelers agreement

Figure 3.12.: Most common words for "What" comments

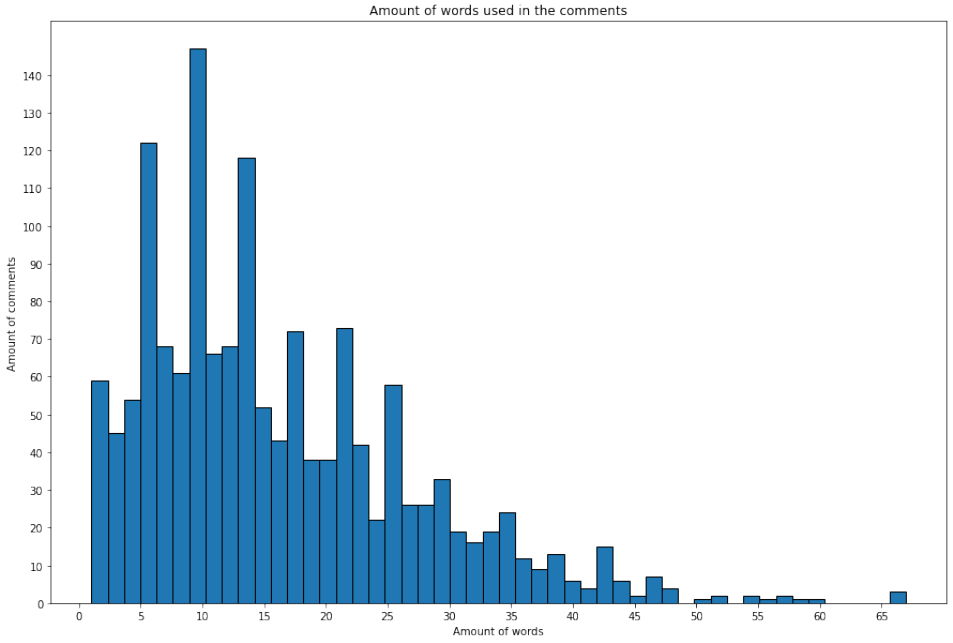


Figure 3.13.: Amount of words used for "What" comments

4. Benchmarking

To assess the quality of semantic similarity in news using current Natural Language Processing models, we conduct two types of benchmarking: **embeddings-based** and **prompting-based**. We formulate the task of measuring News Semantic Similarity as a classification problem, focusing on answers to WHO, WHERE, WHEN, and WHAT questions. We do not consider the generated textual justifications of the WHAT category label.

As ground truth labels, we use those aggregated from overlapping data, excluding any instances of disagreement on the label within a category. Since this is a multi-classification task, we evaluate the models using standard classification metrics: **Accuracy**, **Precision**, **Recall**, and **F1-Score**.

4.1. Embeddings

To our knowledge, there are no designated models for multilingual semantic similarity measurement of news articles covering the Ukrainian language. Therefore, we directly utilize multilingual embedding models to generate (contextualized) embeddings of news article summaries structured around answering the WHO, WHERE, WHEN, and WHAT questions. We rely on summarizations because embedding models, such as BERT-based models, often have a context window limitation (e.g., 512 tokens), which is generally insufficient for news articles' full texts. Modern NLP techniques have demonstrated that summarizations generated by Large Language Models, such as, in our case, the Command R model from Cohere, are of high quality, allowing for a fair benchmarking process in this setting.

To ensure broad coverage, we selected various open-source models, ranging from basic to state-of-the-art, for our benchmark. We use **cosine similarity** to evaluate the semantic similarity between news article embeddings as the gold standard metric of similarity measurement in the field.

The same algorithm for calculating semantic similarity between news articles was applied to all models in this section. First, we divide the summarizations of news articles generated by the Command R model (as explained in Subsection 3.1.2) into four parts, corresponding to answers to the WHO, WHERE, WHEN, and WHAT questions. This division is straightforward due to the structure of the generated summaries.

Next, we generate embeddings for each of these four parts for each article in a pair. We then compute the cosine similarity between the corresponding parts of the two articles in each pair and visualize the distribution of cosine similarities using boxplots. These boxplots additionally reflect key statistical measures such as mean, maximum, minimum, and standard deviation for each question group.

To convert the task into a classification problem, we identify thresholds for each model and question part based on the distributions, allowing us to compare our results with ground truth labels. These thresholds are based on the assumption that higher cosine similarity values indicate that the news articles are more closely related, so we assume that these thresholds could divide the cosine similarity distribution into "Different," "Somewhat related," and "Similar" categories of news article pairs. We removed pairs of news articles labelled as "incomparable" since deciding which threshold could identify this class was untrivial.

Category	Train	Test
Who	82	400
When	80	300
Where	86	370
What	93	400

Table 4.1.: Train and Test Split

We randomly split the remaining dataset into train and test sets, however, ensuring that both sets contain all classes. The train and test data samples distribution for each part are shown in Table 4.1.

We then use a **decision tree classifier**[1] from the Sklearn Python library to learn the thresholds for each model and each question. We limit the maximum depth of the decision tree to two, and class weights are adjusted automatically, inversely proportional to class frequencies. We apply the learned thresholds to the test data and compute multi-classification metrics: overall accuracy, as well as class-wise accuracy, F1-score, recall, and precision.

The results for each model are discussed in the following subsections.

4.1.1. Bag-of-Words

The Bag-of-Words (BoW) model is a simple method for representing text data without taking semantic similarity into account. Despite its inability to capture semantics, it remains widely used for similarity estimation in Natural Language Processing (NLP) due to its simplicity, language and domain independence, and explainability. In the Bag-of-Words approach, a document is viewed as a collection of word frequencies where word order is ignored. Each document in the corpus is represented as a vector, with the vector's length equal to the vocabulary size. Each element in the vector represents the frequency of a specific word appearing in the document.

Due to its implementation design, the Bag-of-Words model cannot be directly applied for cross-lingual similarity measurement. Therefore, we translated the Russian, Polish, and English summarization parts into Ukrainian using DeepL. After translation, we implemented the BoW model using the **CountVectorizer**[2] class from the scikit-learn Python library. This class tokenizes the text and generates a matrix of word counts, where each row represents a

[1] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

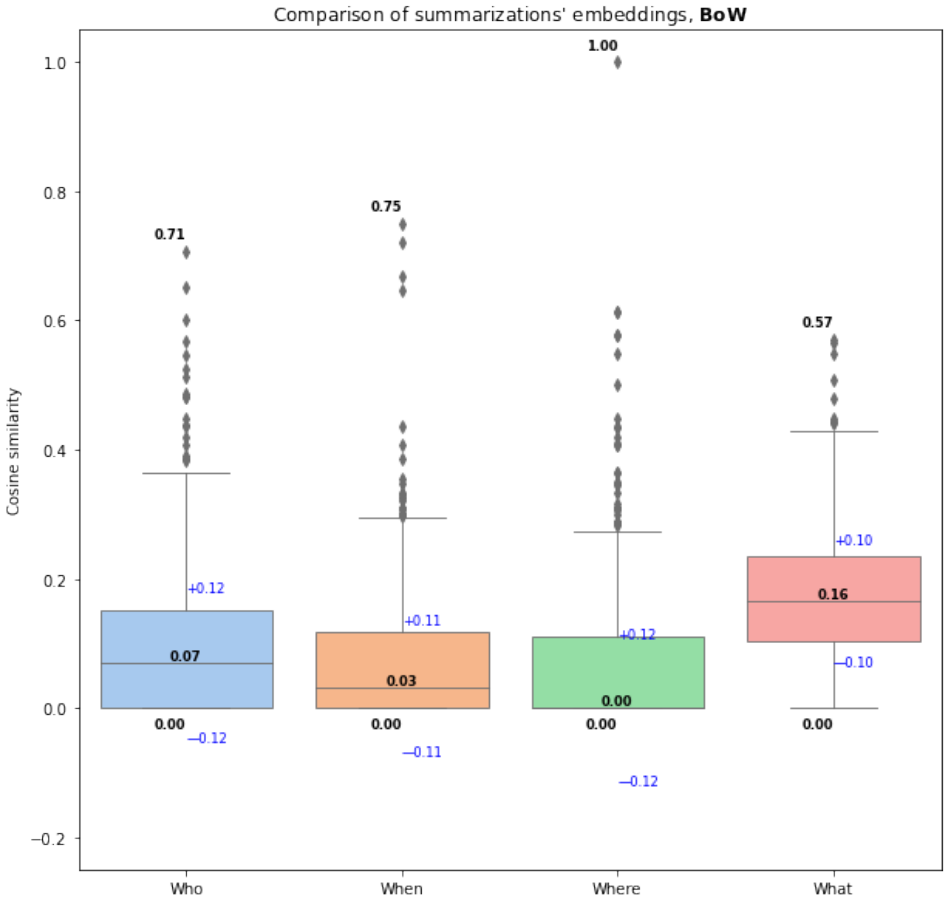


Figure 4.1.: Cosine similarity distribution, Bag-of-Words

document, and each column corresponds to a word from the vocabulary. The distribution of Cosine Similarity using this model is shown in Figure 4.1.

The thresholds that the decision tree learned based on the distribution for WHO, WHEN, WHERE, and WHAT categories are presented in Figure A.1.

Category	**Accuracy**
Who	0.44
When	0.83
Where	0.54
What	0.63

Table 4.2.: Overall Accuracy, Bag-of-Words, Cosine Similarity

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.25	0.88	0.15
	Partly	0.18	0.20	0.16
	No	0.61	0.46	0.9
When	Yes	0.2	0.27	0.15
	Within One Month	0	0	0
	No	0.92	0.89	0.94
Where	Yes	0.43	0.54	0.35
	Partly	0.1	0.06	0.27
	No	0.71	0.79	0.65
What	Similar	0.34	0.5	0.26
	Somewhat Related	0.4	0.42	0.38
	Different	0.76	0.72	0.81

Table 4.3.: Classification metrics, Bag-of-Words, Cosine Similarity

We evaluated the Bag-of-Words model on the test set using the learned thresholds, achieving the overall accuracy presented in Table 4.2. Class-wise classification metrics are reported in Table 4.3. Surprisingly, the model performs well, judging by the "Overall accuracy" metric. However, it might be due to class imbalance since the majority of elements lie in the class "no" or "different", so the model which predicts the negative class statically will have a high overall accuracy. So, we need a detailed comparison of the F1-score, a class-balanced metric, to draw conclusions.

4.1.2. BERT

Bidirectional encoder representations from transformers (BERT) [22] is a commonly used corpus-based model for Semantic Similarity measurement. BERT captures the context of

words bidirectionally (on both sides), which is important for semantic similarity, where the model must grasp the overall meaning of entire sentences or texts.

For this study, we used the multilingual 179M parameters version of BERT[3] with a help of the HuggingFace **transformers** library[4]. During pretraining, it focuses on two main objectives: Masked Language Modeling (MLM), where random words in the text are masked, and the model learns to predict them, and Next Sentence Prediction (NSP), where the model predicts whether one sentence follows another in a given text. The **bert-base-multilingual-cased** model was trained on large multilingual Wikipedia datasets covering 104 languages. Languages with larger Wikipedias were under-sampled, while low-resource languages were oversampled to ensure balanced representation.

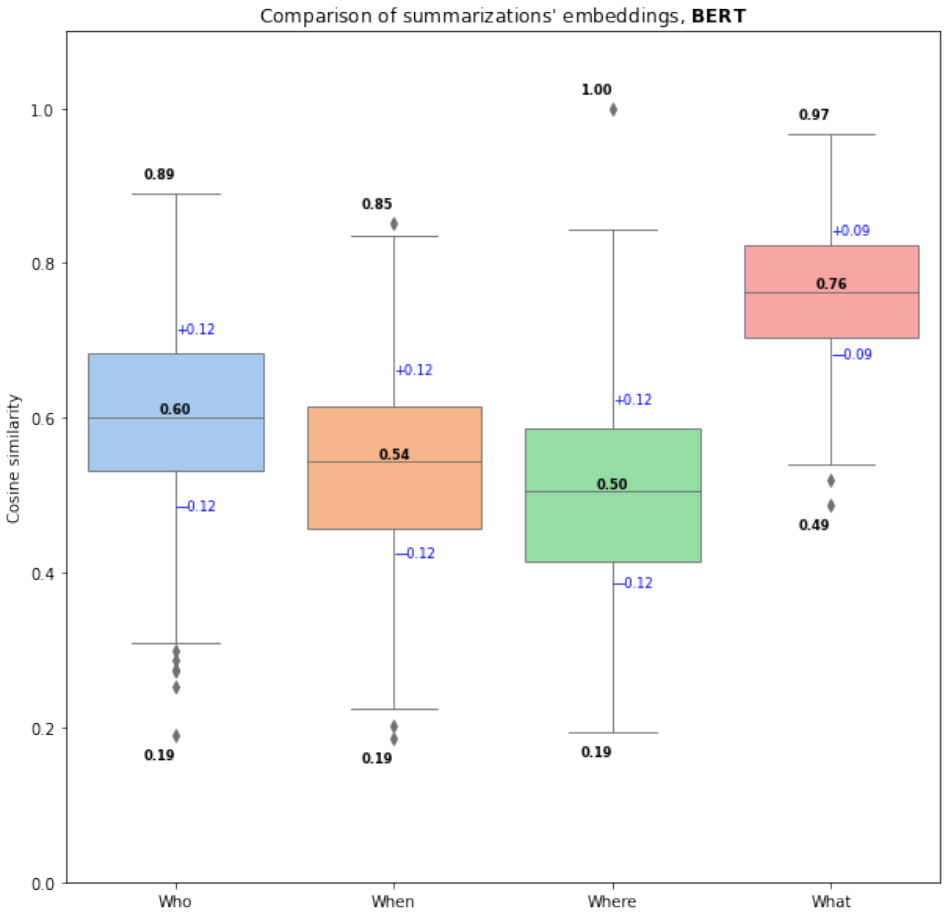


Figure 4.2.: Cosine similarity distribution, BERT

Figure 4.2 shows the distribution of Cosine Similarity scores produced by BERT. Based on this distribution, Figure A.2 presents the thresholds learned by the decision tree for each question category: WHO, WHEN, WHERE, and WHAT.

[3]https://huggingface.co/google-bert/bert-base-multilingual-cased
[4]https://huggingface.co/docs/transformers/en/index

Category	Accuracy
Who	0.3
When	0.53
Where	0.43
What	0.24

Table 4.4.: Overall Accuracy, BERT, Cosine Similarity

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.31	0.28	0.35
	Partly	0.34	0.88	0.21
	No	0.26	0.15	0.9
When	Yes	0.71	0.56	0.95
	In One Month	0.04	0.1	0.02
	No	0.71	0.56	0.95
Where	Yes	0.31	0.43	0.24
	Partly	0.03	0.02	0.08
	No	0.6	0.63	0.57
What	Similar	0.08	0.5	0.04
	Somewhat Related	0.03	0.02	0.06
	Different	0.41	0.31	0.62

Table 4.5.: Classification metrics, BERT, Cosine Similarity

We evaluated the BERT model on the test set using the learned thresholds, achieving the overall accuracy presented in Table 4.4. Per class classification metrics are reported in Table 4.5. The results were generally worse compared to the Bag-of-Words model in "Overall Accuracy"; however, they were more balanced per class in the F1-score for categories WHO and WHEN. In general, low performance might be due to the BERT model's case sensitivity or the fact that the BERT model was pre-trained but not fine-tuned specifically for semantic similarity tasks.

4.1.3. mt5-small

The Text-to-Text Transfer Transformer (T5) is an encoder-decoder model based on the idea that all NLP tasks can be framed as a "text-to-text" problem: instruction in, text results out. This unified approach allows the model to generalize across a wide range of tasks, including semantic similarity. T5 is pre-trained on a masked language modelling "span-corruption" objective. In this objective, consecutive spans of input tokens are replaced with a mask token, which the model is trained to reconstruct.

Our work used the multilingual version of T5, known as mT5, which was pre-trained on the Multilingual Colossal Clean Crawled Corpus (mC4) [78]. This dataset includes 101 languages, covering Ukrainian, Russian, Polish, and English. The model is available in different sizes up to 13B parameters. We selected the **mT5-small** model[5], with 300M parameters, due to our resource constraints: this model is not accessible via the Hugging Face inference API[6] and must be run locally using the Hugging Face transformers library.

Figure 4.3 shows the distribution of Cosine Similarity scores using the mt5-small model. Based on this distribution, Figure A.3 presents the thresholds learned by the decision tree for each category: WHO, WHEN, WHERE, and WHAT.

Category	Accuracy
Who	0.62
When	0.66
Where	0.41
What	0.49

Table 4.6.: Overall Accuracy, mt5-small, Cosine Similarity

We evaluated the mt5-small model on the test set using the learned thresholds, with the resulting overall accuracy presented in Table 4.6. Multi-classification metrics are reported in Table 4.7. Due to its generalization abilities coming from the text-to-text paradigm, mt5-small performs significantly better than BERT in terms of "Overall Accuracy" (also, its size being nearly twice as large surely affects it). However, its high overall accuracy in the category WHEN does not reflect much due to the noticeable tendency of model to predict the negative class for this category.

[5]https://huggingface.co/google/mt5-small
[6]https://huggingface.co/docs/api-inference/en/index

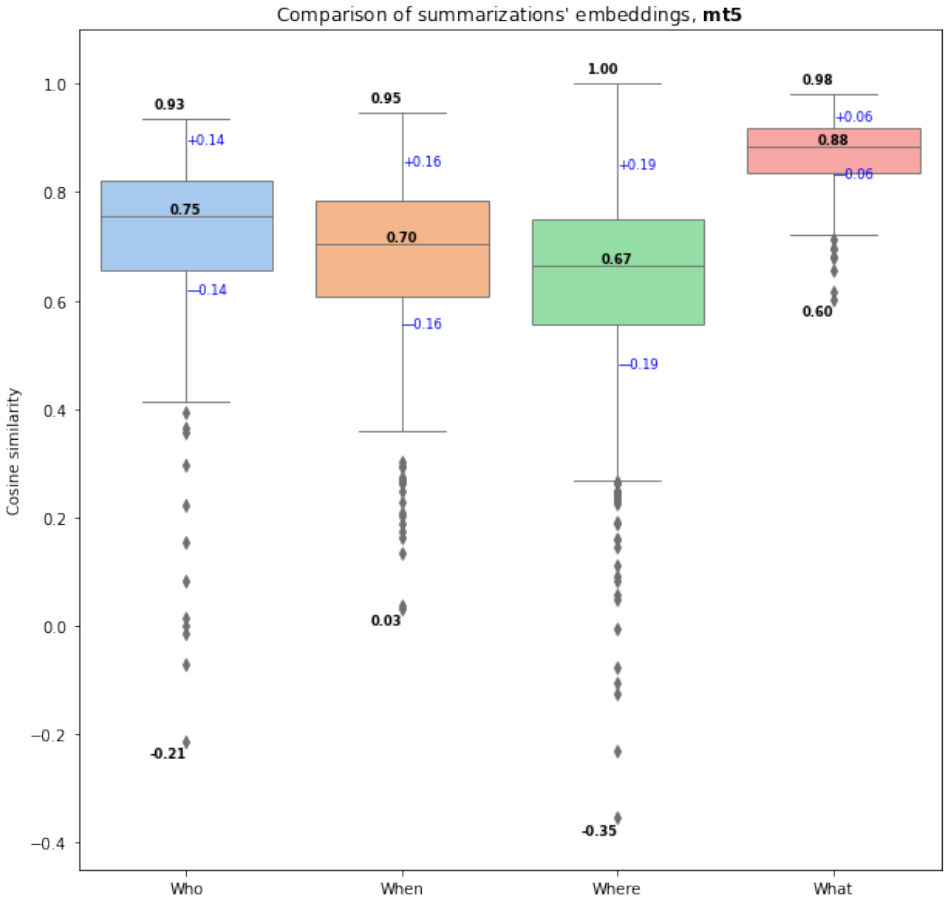


Figure 4.3.: Cosine similarity distribution, mt5-small

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.23	0.28	0.2
	Partly	0.26	0.29	0.24
	No	0.77	0.74	0.8
When	Yes	0.02	0.07	0.01
	In One Month	0	0	0
	No	0.8	0.72	0.9
Where	Yes	0.27	0.44	0.2
	Partly	0	0	0
	No	0.58	0.61	0.56
What	Similar	0.23	0.7	0.14
	Somewhat Related	0.19	0.18	0.21
	Different	0.68	0.59	0.79

Table 4.7.: Classification metrics, mt5-small, Cosine Similarity

4.1.4. XLM-RoBERTa

We used **xlm-roberta-base**[7] from Facebook AI, a multilingual model with 279 million parameters, based on the Robustly Optimized BERT Pretraining Approach (RoBERTa) [79]. RoBERTa is an improved version of BERT trained on a larger dataset (about ten times bigger) for a more extended period of time. Unlike BERT, which uses a static masking technique (where the same approach for selecting and masking tokens is used on each training epoch), RoBERTa employs dynamic masking, which makes the model more robust. XLM-RoBERTa was pre-trained on 2.5TB of filtered CommonCrawl data, which includes text from 100 different languages. The model was trained in a self-supervised manner, without human-labeled data, using raw text only.

Figure 4.4 depicts the distribution of Cosine Similarity scores estimated by XLM-RoBERTa. Based on this distribution, Figure A.4 presents the thresholds found by the decision tree for each category: WHO, WHEN, WHERE, and WHAT.

Category	**Accuracy**
Who	0.37
When	0.2
Where	0.46
What	0.15

Table 4.8.: Overall Accuracy, XLM-RoBERTa, Cosine Similarity

[7] https://huggingface.co/FacebookAI/xlm-roberta-base

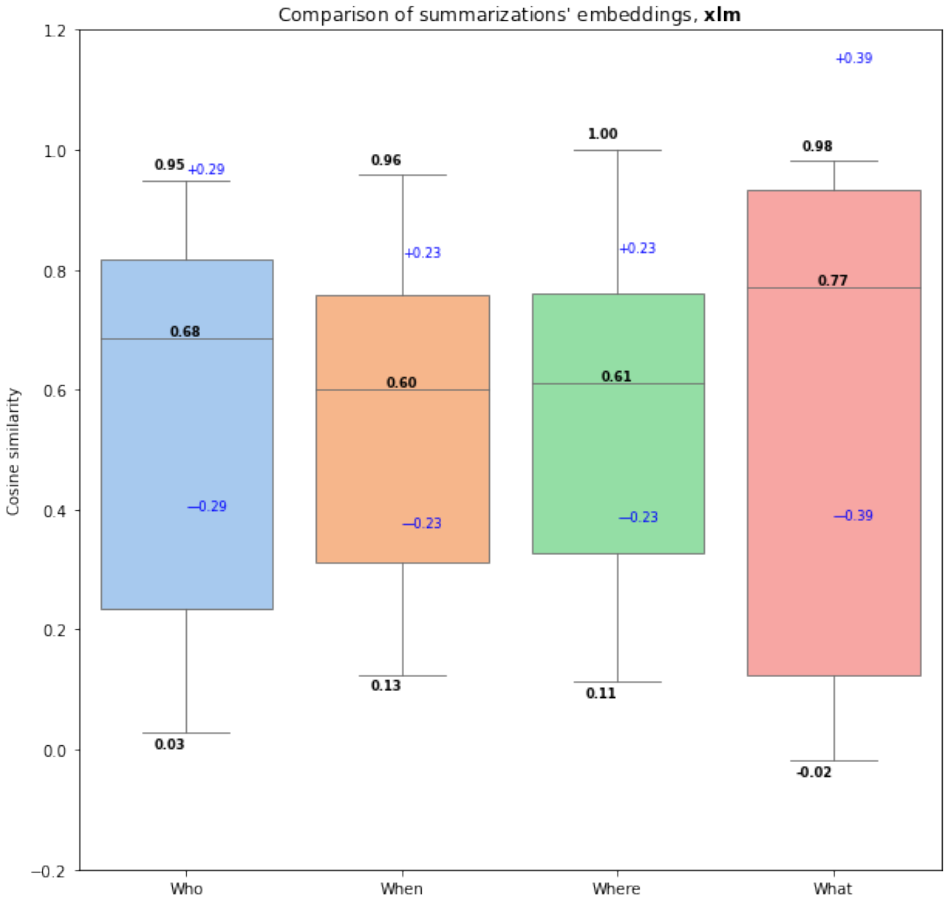


Figure 4.4.: Cosine similarity distribution, XLM-RoBERTa

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.17	0.4	0.11
	Partly	0.26	0.41	0.19
	No	0.49	0.36	0.78
When	Yes	0.1	0.87	0.05
	In One Month	0	0	0
	No	0.3	0.17	0.96
Where	Yes	0.29	0.31	0.27
	Partly	0.15	0.11	0.26
	No	0.62	0.69	0.56
What	Similar	0.14	0.9	0.08
	Somewhat Related	0.3	0.38	0.25
	Different	0	0	0

Table 4.9.: Classification metrics, XLM-RoBERTa, Cosine Similarity

We evaluated the XLM-RoBERTa model on the test set, achieving the overall accuracy presented in Table 4.8. Additional classification metrics are reported in Table 4.9. It seems to us counter-intuitive that XLM-RoBERTa shows worse results than BERT compared by overall and class-wise metrics.

4.1.5. Multilingual e5-large

We used the multilingual **e5-large model**[8] via the Hugging Face API. This model is based on **xlm-roberta-large** (561M parameters version of XLM RoBERTa mentioned in the previous subsection). It supports 100 languages, though it is reported that performance may decrease for low-resource languages, making its accuracy in Ukrainian uncertain for us. Since the model is primarily designed for retrieval tasks, it should perform well in semantic similarity tasks.

The model's training process follows a two-stage approach: first, weakly supervised contrastive pre-training on billions of text pairs, followed by supervised fine-tuning on a smaller set of high-quality labelled data [80]. It has been trained on datasets such as mC4, Wikipedia, Multilingual CC News (particularly relevant for our use case), Reddit, and others. According to the model card, the cosine similarity scores generated by this model tend to fall between 0.7 and 1.0 due to the use of a low temperature (0.01) in the InfoNCE contrastive loss function.

Figure 4.5 shows the distribution of Cosine Similarity scores using this model, demonstrating the predicted in the model card distribution range. Figure A.5 presents the thresholds learned by the decision tree based on the cosine similarity distribution for each category:

[8]https://huggingface.co/intfloat/multilingual-e5-large

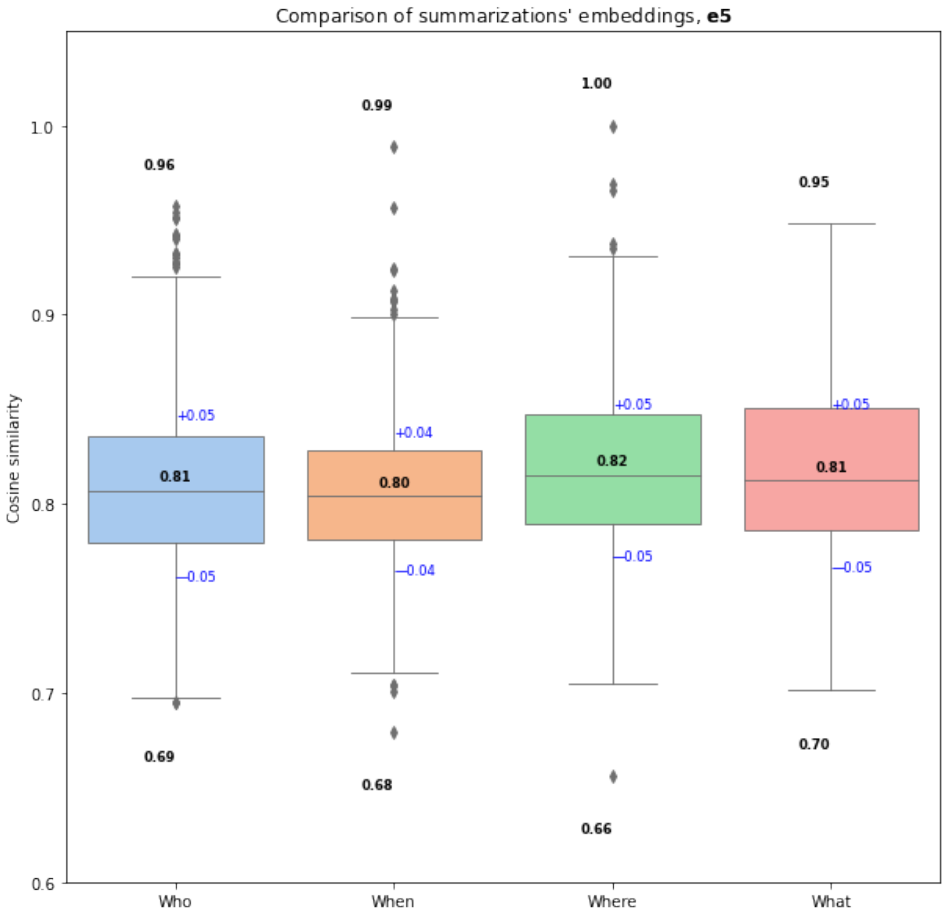


Figure 4.5.: Cosine similarity distribution, multilingual e5-large

WHO, WHEN, WHERE, and WHAT.

Category	Accuracy
Who	0.65
When	0.42
Where	0.5
What	0.57

Table 4.10.: Overall Accuracy, multilingual e5-large, Cosine Similarity

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.44	0.76	0.31
	Partly	0.34	0.41	0.29
	No	0.79	0.7	0.91
When	Yes	0.13	0.87	0.07
	In One Month	0.14	0.1	0.25
	No	0.58	0.41	0.96
Where	Yes	0.52	0.71	0.41
	Partly	0.38	0.45	0.32
	No	0.58	0.46	0.8
What	Similar	0.64	0.85	0.52
	Somewhat Related	0.46	0.7	0.35
	Different	0.64	0.5	0.88

Table 4.11.: Classification metrics, multilingual e5-large, Cosine Similarity

We evaluated the multilingual e5-large model on the test set using the learned thresholds, with the resulting overall accuracy presented in Table 4.10. Per class classification, metrics are reported in Table 4.11. As expected, the model has, in general, better results than all the models. It's well-balanced in its predictions, which negatively affects its overall accuracy compared to Bag-of-Words, but class-wise, it shows superior performance.

It is extremely interesting why such a simple model - Bag-of-Words - without a semantical "understanding" performs not so much worse than the models which are designed to fight the semantical gap. It might be since some of them weren't downstreamed to the task of semantic similarity measurement, and none of them were fine-tuned to measure news semantic similarity. Additionally, due to the Ukrainian language being a low-resourced one, their performance might suffer on it, while Bag-of-Word is language and domain-agnostic.

4.2. Large Language Models

Large language models (LLMs) benchmarked in this section are not suited for encoding tasks due to their architecture, which is decoder-only. Recent studies, like the one by Behnamghader et al. (2024) [81], show that it is not impossible to adapt decoder-only models for encoding tasks. It can be done through fine-tuning through the following steps: 1) enabling bidirectional attention, 2) implementing masked next-token prediction, and 3) applying unsupervised contrastive learning. It will be interesting to see whether this method becomes widely adopted, as it involves adapting autoregressive models for the task for which they were not pretrained. Currently, we do not consider this method feasible, so the only way to utilize decoder-only LLMs in our context is through prompting.

To provide more context to models, we use both versions of summarizations within the prompt. Here, we treat summarization as equivalent to chain-of-thought prompting, where the first step involves extracting the required information from news sources. To enhance the models' performance, we use one-shot prompting. The models are instructed to produce their output in a JSON format similar to the output received from the Toloka platform after labelling. We benchmark models on the whole dataset, excluding "disagreement" labels. Using prompting for evaluation, we can additionally check models' ability to classify "incomparable" pairs, which is much less trivial for embedding-based benchmarks.

The following is the template one-shot prompt we constructed based on the crowdsourcing project instructions. We adapt this prompt based on the best prompting practices for each model, referenced in the following sections.

Template One-Shot Prompt for LLMs Benchmarking

```
Your goal is to compare news articles.
For each news article, two summarization options are provided
(including basic information on four parameters: WHO, WHEN, WHERE and
WHAT happened). News in a pair can be in different languages, but this
should not affect your comparison. For example, Wołodymyr Zełenski
(Polish) and Volodymyr Zelenskyy (English) are the same actors in an
event. You have to answer 4 sub-questions about the similarity of news
in JSON format:

**Do the main characters in the news match? (WHO?)**
The actor of an article can be a specific person (Volodymyr Zelenskyy),
an organization/party or any other association of people.
1. **yes**. The main and secondary protagonists in the articles are the
same. At the same time, it is important to note that one actor can be
called differently, for example,
"Volodymyr Zelenskyi" and "the President of Ukraine".
2. **partly**. (There are overlaps) Some of the main or secondary actors
are the same.
```

```
3. **no**. The actors in the articles do not overlap at all;
they have nothing in common.
4. **incomparable**. At least one article has no actors. For example,
one of the articles states something very vague, like
‘‘the entire population of the Earth’’ or the article is a weather forecast.

**Do the places where the events in the news are taking place match
the places in the news? (WHERE?)**
The same place may be referred to in different ways, such as
"St. Petersburg" and "Peter".
1. **yes**. Articles mention exactly the same place. That is, if one
article mentions a specific place/city/country, it should be mentioned
in the second article. If one article mentions only a country and the
other - a city, this is not an exact match.
2. **partly**. (There are overlaps) The place mentioned in one article is
located within the boundaries of the place mentioned in another article.
For example, if one article refers to events taking place in Ukraine
and the other in Kyiv, this option is selected. It is also a valid choice
if different places (e.g., countries) are mentioned in the articles and
some of the names of the places are the same.
3. **no**. The places are completely different (different places
in the city, in the country, different countries).
4. **incomparable**. At least one article does not mention the location
at all. Something very vague, for example, something that happens all
over the world, all over the Internet, or if the location is not really
specified. At the same time, if the place of action can be understood,
for example, the article says "in our country," and the author is
obviously a Ukrainian, then this option should not be selected.

**Does the timing of the events in the news coincide? (WHEN?)**
It is important to understand that we are interested in the time/date
of the news event, NOT the time/date of the article’s publication.
1. **yes**. Same event time-wise. It is obvious from the article that
it is about the same event that happened at the same time.
2. **partly**. Two events in news articles occurred within the same month.
3. **no**. The difference between the times of the events is more than
a month.
4. **incomparable**. From at least one news article, it is impossible to
establish the time of the event; it is impossible to determine when
the event itself took place or the time period in the article is very
vague, for example, ‘‘at all times’’.
```

```
**How do the news topics/events relate to each other? (WHAT?)**
1. **similar**. News articles are about the same event.
2. **somewhat_related**. Common specific topic/common event.
3. **different** News are about different events.
1. **similar**
News articles describe the same event that happened in the same place,
with the same people at approximately the same time.
However, the event might be described:
- with different focus (one focuses more on one part of the event,
the other - on another)
- from different points of view, political positions
- with errors in the details, which makes the news seem ‘‘different’’
If you are not 100% sure that the news is about the same event
with the same actors, in the same place and at approximately the same
time (some details make them different), then you should choose the
option ‘‘Common specific topic/common event’’.
2. **somewhat_related**
News articles are related to one close, narrow topic/event but describe
different incidents within it.
Examples of narrow common topics:
- Curfews imposed due to COVID-19 (in different countries or changes in
its status within the same country);
- Robbery of a grocery store by teenagers (in different places or
different events within the same case);
- Joe Biden’s 2020 election campaign and various events described from it.
News common topics should be narrow and specific: for example, two news
stories on broad and popular topics in the news, aka COVID, death, and
elections do not necessarily have to have any commonality in themselves.
3. **different**
Events in the news have little to do with each other. They either do not
have a common theme, or the common theme is so broad (e.g., COVID) that
it is not a reason to consider the news similar.

Provide **a short reasoning** for **WHAT?** answer in the following
format:
- News are **similar**: Briefly describe, summarizing in 1-2 sentences,
what the event described in both news is about.
- Both news are **somewhat_related**: Briefly, in 1-2 sentences, identify
the common specific topic of news.
- News articles are about **different** events: Briefly summarize in
1-2 sentences: what are the first and the second news articles about?
```

```
Input format example:
**Summarization of the first news article, version 1**


ХТО: 27 учнів середньої школи з Марбурга та їхній водій.
КОЛИ: В неділю, 2023.
ДЕ: На автобані 45 у німецькій федеральній землі Саар.
ЩО: Двоповерховий автобус, який віз 27 учнів середньої школи та їхніх
вчителів в Англію, потрапив у серйозну аварію. Три дівчини та хлопець
отримали серйозні травми і були госпіталізовані, а також водій автобуса
отримав легкі травми. Автобус з'їхав з дороги та перекинувся на бік.
Поліція розслідує причину ДТП. Через аварію автострада у напрямку
Дортмунда була повністю блокована. Шкільна екскурсія була скасована,
і дітей повернули додому. Це не перша така аварія за участі шкільного
автобуса в Німеччині - у квітні та грудні минулого року також були
серйозні ДТП за участі шкільних автобусів, у яких постраждали діти.


 **Summarization of the first news article, version 2**


ХТО: 27 учнів середньої школи у Марбурзі та їх 54-річний водій автобуса.
КОЛИ: У неділю о 6:27, 2023.
ДЕ: Автобан 45 у Саарі, неподалік Вендена в районі Ольпе.
ЩО: Двоповерховий автобус, який віз 27 учнів та п'ятеро вчителів зі
старшої школи з німецького Марбурга в Англію, потрапив у аварію.
За словами поліції Дортмунда, автобус з'їхав у праву сторону та
перекинувся на бік. У результаті аварії постраждали 27 учнів та їх водій,
чотири людини отримали серйозні травми, а решта - легкі. Усі постраждалі
були доставлені до лікарні, їхньому життю нічого не загрожує.
Шкільна екскурсія була скасована, а дітей повернули до Марбурга.
Через аварію автобан у напрямку Дортмунда був повністю закритий протягом
дня. Ця подія є не єдиною аварією шкільного автобуса в Німеччині. Раніше
у квітні в місті Гольцмінден аварія за участю шкільного автобуса призвела
до госпіталізації 16 учнів, а в грудні минулого року - до загибелі дитини
через сильну ожеледицю.


 **Summarization of the second news article, version 1**


ХТО: Стрілянину в Огайо влаштував 27-річний чоловік, через його дії одна
людина загинула, а 26 отримали поранення. А перед цим у Словаччині було
```

```
скоєно замах на прем'єра Роберта Фіцо, а в Туреччині - на відділок
поліції, де загинули двоє людей.
КОЛИ: Масштабна стрілянина в Огайо відбулася в ніч на неділю, 2 червня,
2024. Замах на Роберта Фіцо відбувся 15 травня, а стрілянина в Туреччині
- у відділенні поліції - сталася невдовзі після цього.
ДЕ: Події розгорталися в американському штаті Огайо, specifically на
вулиці міста Акрон. У Словаччині замах на прем'єра відбувся перед
місцевим Будинком культури в місті Хандлов.
А в Туреччині - у місті Адияман, у відділку поліції.
ЩО: В результаті стрілянини в Огайо одна особа загинула, а 26 отримали
поранення; поліція прибала на місце події і виявила зброю. Це вже другий
за останній час інцидент зі стріляниною в цьому штаті. Замах на прем'єра
Словаччини залишив його у важкому стані, а в результаті стрілянини в
Туреччині загинули двоє правоохоронців. Обидві події показують зростання
насильства з використанням зброї у цих регіонах.


 **Summarization of the second news article, version 2**


ХТО: Нападником виявився 27-річний чоловік, який відкрив стрілянину і
вбив одну людину, а ще поранив 26 людей. Прем'єр Роберт Фіцо.
КОЛИ: Стрілянина відбулася опівночі у неділю, 2 червня, 2024. А замах на
політика Роберта Фіцо стався 15 травня.
ДЕ: Інцидент із стріляниною відбувся у штаті Огайо, на вулиці міста Акрон.
А напад на Роберта Фіцо відбувся у місті Хандлов.
ЩО: У штаті Огайо відбулася стрілянина, за участі 27-річного стрільця,
внаслідок якої 1 особа загинула та ще 26 отримали поранення різного
ступеня тяжкості. Правоохоронці вилучили зброю на місці події. Окрім того,
у своїй відповіді я розповіла про замах на прем'єра Словаччини Роберта
Фіцо, який відбувся 15 травня та про стрілянину у відділку поліції в
турецькому місті Адияман, де правоохоронець відкрив вогонь по колегах,
в результаті чого загинуло двоє людей та восьмеро було поранено.


Output format example:
{
"WHO": "no",
"WHO_reasoning": "German schoolchildren vs shooter from Ohio",
"WHERE": "no",
"WHERE_reasoning": "Saar vs Ohio",
"WHEN": "no",
"WHEN_reasoning": "2nd of June, 2024, and any event in 2023 are more than
```

```
a month apart",
"WHAT": "different",
"WHAT_reasoning": "The bus with schoolchildren that flipped over in
Saarland, Germany, has nothing to do with the shooting in Ohio,
which resulted in 27 people dead",
}

Input:
**Summarization of the first news article, version 1**
{SUMMARIZATION 1.1}
**Summarization of the first news article, version 2**
{SUMMARIZATION 1.2}
**Summarization of the second news article, version 1**
{SUMMARIZATION 2.1}
**Summarization of the second news article, version 2**
{SUMMARIZATION 2.2}

Return answer in JSON format:
```

4.2.1. Mistral 7B Instruct

Mistral 7B [49] is an open-source, transformer-based large language model. It was developed with the idea that scaling up model size is less feasible for open-source communities than for closed-source proprietary models. Therefore, Mistral AI focuses on delivering high performance while maintaining models of a more manageable size, like Mistral 7B. This is mainly achieved through techniques like Grouped-Query Attention (GQA), where attention is calculated over grouped queries instead of individual ones, and Sliding Window Attention (SWA), which allows each layer to attend to a fixed amount of prior hidden states (4096 for Mistral 7B) while using stacked transformer layers to access information beyond the fixed window size, providing a theoretical attention span of approximately 131K tokens.

For our experiments, we used the 7.25B parameter instruct version of the Mistral 7B model, **Mistral-7B-Instruct-v0.3**[9]. It was fine-tuned using publicly available instruct datasets from Hugging Face. We chose it since instruct models are easier to use in conversational settings, where we provide it with instruction-based prompts. It has a window context size of 8192 tokens, allowing us to fit the guidelines used for the crowdsourcing project. To optimize the model's performance for our task, we followed the advice provided in *"How to prompt Mistral AI models and why?"* article[10] and adjusted our prompt template accordingly.

We evaluated answers of the Mistral 7B Instruct model, resulting in an overall accuracy demonstrated in Table 4.12 and multi-class classification metrics: accuracy, precision, recall,

[9]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[10]https://community.aws/content/2dFNOnLVQRhyrOrMsloofnW0ckZ/how-to-prompt-mistral-ai-models-and-why

Category	Accuracy
Who	0.24
When	0.09
Where	0.27
What	0.41

Table 4.12.: Overall Accuracy, Mistral 7B Instruct, Cosine Similarity

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.26	0.53	0.17
	Partly	0.35	0.77	0.23
	No	0.09	0.05	0.89
	Incomparable	0.1	0.2	0.06
When	Yes	0.12	0.15	0.11
	Within One Month	0.08	0.89	0.04
	No	0.02	0.01	0.6
	Incomparable	0.26	0.18	0.46
Where	Yes	0.38	0.8	0.25
	Partly	0.32	0.36	0.29
	No	0.03	0.02	0.8
	Incomparable	0.32	0.43	0.26
What	Similar	0.19	0.63	0.11
	Somewhat Related	0.29	0.3	0.28
	Different	0.56	0.44	0.75

Table 4.13.: Classification metrics, Mistral 7B Instruct, Cosine Similarity

and F1-Score showed in Table 4.13. The quality of results is low, lower than most of the embedding models. It was not so surprising, considering the quality of the summarization of news articles we encountered while selecting a candidate model for news article summarization. It might be due to the fact that the Ukrainian language is low-resourced or that the task of measuring semantic news similarity is too complicated.

4.2.2. Mixtral 8x7B Instruct

We selected another model from Mistral AI, **Mixtral-8x7B-Instruct-v0.1**[11]. Mixtral-8x7B model is a Large Language Model (LLM) based on a sparse mixture of expert models, which, to simplify it extremely, consists of jointly working eight groups of Mistral models. At each layer, for every token, a router network chooses two of these groups to process the token, combining their outputs. While Mixtral has a total of 46.7 billion parameters, it only uses 12.9 billion parameters per token, operating at the speed and cost of a model of the 12.9B size. Mixtral 7x8B outperforms GPT-3.5 on most standard benchmarks, which might promise a decent quality on semantic similarity tasks. Additionally, it features a large 32k context window, four times larger than Mistral 7B one. However, the model is primarily optimized for English, French, Italian, German, and Spanish languages, so its performance in the Ukrainian language remains uncertain.

The Mixtral-8x7B Instruct model is optimized for instruction following through supervised fine-tuning and direct preference optimization (DPO). We chose the instruct model for reasons similar to those mentioned in the previous section. We also applied the same techniques for designing the prompts as we did for the Mistral 7B Instruct model.

Category	Accuracy
Who	0.22
When	0.48
Where	0.38
What	0.67

Table 4.14.: Overall Accuracy, Mixtral 8x7B Instruct, Cosine Similarity

We evaluated answers provided by the Mixtral 8x7B Instruct model, receiving overall accuracy demonstrated in Table 4.14 and multiclass classification metrics showed in Table 4.15. The results show a clear improvement compared to the Mistral 7B model, particularly in the WHAT category, where it outperforms all embedding models. However, in the WHO, the model performs extremely low when it comes to predicting negative class, which is surprising.

[11] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.37	0.59	0.26
	Partly	0.33	0.8	0.2
	No	0.01	0.01	1.0
	Incomparable	0.29	0.2	0.5
When	Yes	0.18	0.45	0.11
	Within One Month	0.18	0.26	0.13
	No	0.61	0.47	0.84
	Incomparable	0.41	0.55	0.33
Where	Yes	0.52	0.74	0.4
	Partly	0.16	0.1	0.43
	No	0.47	0.37	0.66
	Incomparable	0.2	0.78	0.12
What	Similar	0.5	0.67	0.4
	Somewhat Related	0.44	0.46	0.42
	Different	0.79	0.75	0.82

Table 4.15.: Classification metrics, Mixtral 8x7B Instruct, Cosine Similarity

4.2.3. Llama 3.1 8B Instruct

Llama 3.1, the latest model in Meta AI's Large Language Model (Llama) family, was released on 23 July 2024. It is a decoder-only autoregressive language model built using an optimized through Grouped-Query Attention transformer architecture. The Llama 3.1 Instruct version is enhanced through supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF), aligning it with human preferences. With a context length of 128k tokens, it handles significantly larger inputs than all the models in this section.

Llama 3.1 is considered multilingual, with over 5% of its pretraining data in more than thirty non-English languages. However, the explicitly supported languages—English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai—do not include Ukrainian. The model was pretrained on approximately 15 trillion tokens from publicly available sources and fine-tuned on publicly available instruction datasets and over 25 million synthetically generated examples. Apart from standard benchmarks, it was evaluated on a high-quality human evaluation set with 1,800 prompts covering twelve use cases: asking for advice, brainstorming, classification, closed question answering, coding, creative writing, extraction, inhabiting a character/persona, open question answering, reasoning, rewriting, and summarization. However, since the training data is not fully disclosed, it is unclear how well Llama 3.1 Instruct could perform on the news semantic similarity task.

We chose the Llama 3.1 8B Instruct model (**Meta-Llama-3.1-8B-Instruct**[12]) to ensure the comparability of results - model's size-wise - with other models in this section. We adapted

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

the template prompt to Llama 3.1 Instruct format by following Meta's official guide[13].

Category	Accuracy
Who	0.49
When	0.73
Where	0.52
What	0.68

Table 4.16.: Overall Accuracy, Llama 3.1 8B Instruct, Cosine Similarity

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.27	0.28	0.26
	Partly	0.32	0.49	0.24
	No	0.62	0.51	0.77
	Incomparable	0.13	0.2	0.09
When	Yes	0.15	0.1	0.33
	Within One Month	0.09	0.05	0.25
	No	0.83	0.89	0.77
	Incomparable	0.5	0.44	0.57
Where	Yes	0.48	0.65	0.38
	Partly	0.15	0.09	0.38
	No	0.66	0.7	0.63
	Incomparable	0.46	0.52	0.41
What	Similar	0.34	0.22	0.75
	Somewhat Related	0.4	0.38	0.42
	Different	0.79	0.83	0.76

Table 4.17.: Classification metrics, Llama 3.1 8B Instruct, Cosine Similarity

We evaluated the performance of the Llama 3.1 8B Instruct model, with overall accuracy shown in Table 4.16 and multiclass classification metrics presented in Table 4.17. The model demonstrated one of the best-balanced results so far. Compared to embedding models, it has the additional capability of predicting the "incomparable" class. When it comes to selecting a candidate for fine-tuning on the multilingual news similarity dataset, Llama 3.1 Instruct appears to be one of the most promising options.

[13]https://llama.meta.com/docs/model-cards-and-prompt-formats/llama3_1/

4.2.4. Aya-23 8B

We selected the Aya-23 model [82], an open multilingual model instruct fine-tuned (IFT) from Cohere's Command family of decoder-only transformers. Aya-23 builds on the experience of Cohere's earlier model, Aya-101, which used the mt5 encoder-decoder architecture as a backbone. Aya-101 emerged from the need for LLMs that perform well on low-resourced languages. To address this, Cohere created and curated several multilingual datasets, synthetic and human-made, for fine-tuning. One of the key datasets was the Aya dataset, which consists of 204,000 human-curated prompt-response pairs in 65 languages written by native speakers.

Aya-23 was developed with the understanding that the mt5 backbone model should be replaced with a more modern option. The new pre-trained model incorporates advanced transformer optimization techniques, such as Parallel Attention, Rotary Positional Embeddings, and Grouped Query Attention. Aya 23 is reported to outperform models like Mistral and Mixtral on several benchmarks covering the languages it supports, which include Arabic, Chinese (both simplified and traditional), Czech, Dutch, **English**, French, German, Greek, Hebrew, Hindi, Indonesian, Italian, Japanese, Korean, Persian, **Polish**, Portuguese, Romanian, **Russian**, Spanish, Turkish, **Ukrainian**, and Vietnamese. Since the Aya model includes all the languages we focus on in this work, we expected it to perform well on our benchmark.

For better comparability of results in this section regarding the models' size, we chose the **aya-23-8B**[14] variant. With an 8k window size, it should be able to process our template prompt without truncation. To secure optimal performance, we adapted our template prompt using the official guidelines provided by Cohere[15].

Category	Accuracy
Who	0.67
When	0.47
Where	0.55
What	0.48

Table 4.18.: Overall Accuracy, Aya-23 8B, Cosine Similarity

We evaluated the performance of the Aya-23 8B model, with overall accuracy displayed in Table 4.18 and the multi-class classification metrics shown in Table 4.19. Compared to the Llama 3.1 model, we did not observe any significant improvements in performance, even though the Llama 3.1 model card does not explicitly mention proficiency in the languages relevant to our study. It has trouble predicting negative classes in WHERE and WHEN categories, dramatically affecting "Overall accuracy" due to the class imbalance.

Category	Class Label	Metric		
		F1-Score	**Recall**	**Precision**
Who	Yes	0.37	0.81	0.24
	Partly	0.51	0.5	0.51
	No	0.81	0.72	0.9
	Incomparable	0	0	0
When	Yes	0.12	0.15	0.11
	Within One Month	0.08	0.89	0.04
	No	0.02	0.01	0.6
	Incomparable	0.26	0.18	0.46
Where	Yes	0.38	0.8	0.25
	Partly	0.32	0.36	0.29
	No	0.03	0.02	0.8
	Incomparable	0.32	0.43	0.26
What	Similar	0.19	0.63	0.11
	Somewhat Related	0.29	0.3	0.28
	Different	0.56	0.44	0.75

Table 4.19.: Classification metrics, Aya-23 8B, Cosine Similarity

4.3. Conslusion

To fairly compare the models, we should use metrics that account for class imbalance. Based on the dataset distribution shown in Figure 3.11, a static model that always selects class label "No" for WHO, WHEN, and WHERE categories and "Different" for WHAT category would achieve approximately **70%** overall accuracy for "WHO", **80%** for "WHEN", **55%** for "WHERE", and **69%** for "WHAT." One of the better metrics for this is the **macro-averaged F1-score**, which calculates the arithmetic mean of the F1 scores for each class. This method gives equal importance to all classes. The final results are displayed in Table 4.20. Since decoder-only, prompt-based models also predict the class "incomparable," it is unfair to directly compare their performance to models used as encoders for benchmarking.

From these results, we can make a few conclusions. First, the most complex category for all models to predict is WHEN. The challenge is understandable because the time of events can often be determined from both the text and the article's publication date. We did not provide publication dates for the models since we instructed the labellers not to use publication dates as the primary source of information. However, we could have found a way to highlight this information in the instructions for the prompt-based methods. Second, the best-performing encoder model is **e5-large**. It makes sense since it is the most modern model among the encoders we benchmarked, and it is widely used in modern NLP applications like Information

[14]https://huggingface.co/CohereForAI/aya-23-8B
[15]https://docs.cohere.com/docs/crafting-effective-prompts

Category	Encoder Model	F1-score	Decoder-only Model	F1-score
WHO	BoW	0.35	Mistral 7B Instruct	0.20
	BERT	0.30	Mixtral 8x7B Instruct	0.25
	mt5-small	0.42	Llama 3.1 8B Instruct	0.34
	XLM-RoBERTa	0.31	Aya-23 8B	**0.42**
	e5-large	**0.52**		
WHEN	BoW	0.37	Mistral 7B Instruct	0.12
	BERT	**0.49**	Mixtral 8x7B Instruct	0.35
	mt5-small	0.27	Llama 3.1 8B Instruct	**0.39**
	XLM-RoBERTa	0.13	Aya-23 8B	0.12
	e5-large	0.28		
WHERE	BoW	0.41	Mistral 7B Instruct	0.26
	BERT	0.31	Mixtral 8x7B Instruct	0.34
	mt5-small	0.28	Llama 3.1 8B Instruct	**0.44**
	XLM-RoBERTa	0.35	Aya-23 8B	0.26
	e5-large	**0.49**		
WHAT	BoW	0.50	Mistral 7B Instruct	0.35
	BERT	0.17	Mixtral 8x7B Instruct	**0.58**
	mt5-small	0.37	Llama 3.1 8B Instruct	0.51
	XLM-RoBERTa	0.15	Aya-23 8B	0.35
	e5-large	**0.58**		

Table 4.20.: Benchmarks, Macro-averaged F1-score

Retrieval. It seems to be the best choice as a base model for future downstreaming to the news semantic similarity measurement task. However, surprisingly, the simple Bag-of-Words model outperforms multilingual semantically aware encoders still commonly used in the field. This might be because Bag-of-Words is domain and language-agnostic, showing us that simple models are still valuable for low-resourced, complicated domains. Third, **Llama 3.1 8B Instruct** turned out to be the best-performing decoder-only model, which is unexpected since Aya-23 8B is supposed to be more suitable for multilingual tasks like ours. Llama 3.1 8B Instruct so far seems like the best candidate for future downstreaming.

Overall, the performance of all these models is average, if not underwhelming. Larger models like Aya-23 35B or Llama 3.1 405B might perform much better on this task, but further benchmarking is needed to confirm this. In conclusion, fine-tuning smaller models for low-resourced languages and specific domains appears to be a promising option. This approach is accessible to the open research community, as opposed to training humongous decoder-only models.

5. Conclusion

5.1. Contributions

This thesis has made several key contributions to the area of multilingual news semantic similarity measurement.

One of the contributions is our research of the current state of the field, which revealed a sparsity of existing solutions, especially for low-resourced languages like Ukrainian. To our knowledge, no datasets or specialized tools currently address this challenge.

We developed a scalable crowdsourcing pipeline to fill this gap and created a high-quality multilingual dataset of 500 news article pairs in Ukrainian, Russian, English, and Polish based on it. These news pairs were labelled for similarity using the 4W method, evaluating **what** happened in the news, **where** and **when** events occurred, and **who** were the key actors. Additionally, the dataset includes textual justifications generated by labellers, providing brief explanations of similarity labels between news events.

Furthermore, we have provided detailed guidelines for designing a crowdsourcing solution for multilingual news semantic similarity labelling, from scraping to final data aggregation, that can be easily adapted to other languages. A notable contribution is our hybrid quality control method, which combines automated and LLM-based techniques with human oversight. We encourage further usage and improvement of this pipeline while we stress the importance of ethical crowdsourcing practices, such as fair pay and clear feedback cycles for labellers.

Another significant contribution of this work is the extensive benchmarking of modern transformer-based models, embedding- and prompting-based, for the multilingual semantic news similarity task. Our findings show that even state-of-the-art open-sourced models struggle to perform high-quality news semantic similarity measurement and require further fine-tuning. Based on our evaluation, we have identified two promising backbone models for downstreaming: the **e5-large** encoder model and the **Llama 3.1 8B Instruct** decoder model.

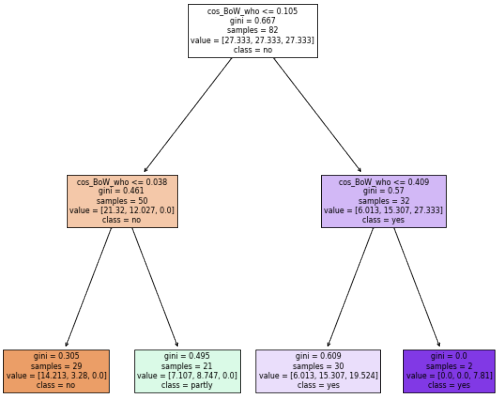
5.2. Limitations and Future Work

Following moderation guidelines on the Toloka platform, we had to exclude news articles covering sensitive topics from our dataset. While this was necessary to protect the mental well-being of Tolokers, it resulted in an underrepresentation of news related to global armed conflicts — topics often central to the spread of fake news and propaganda. Additionally, due to resource limitations, we weren't able to benchmark closed-source models such as GPT-4o and Anthropic's Claude 3, which may have provided further insights into model performance on multilingual news semantic similarity measurement in low-resourced languages. However,

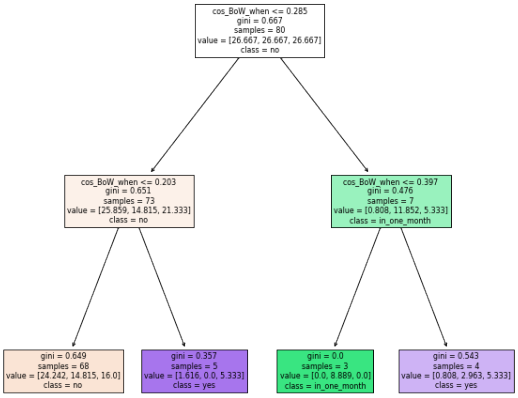
our primary focus was on enhancing open-sourced research.

For future research, we want to explore the impact of fine-tuning the backbone models we identified through benchmarking on our dataset. Fine-tuning could significantly improve the models' ability to handle complex tasks like multilingual news semantic similarity, particularly for low-resourced languages such as Ukrainian.

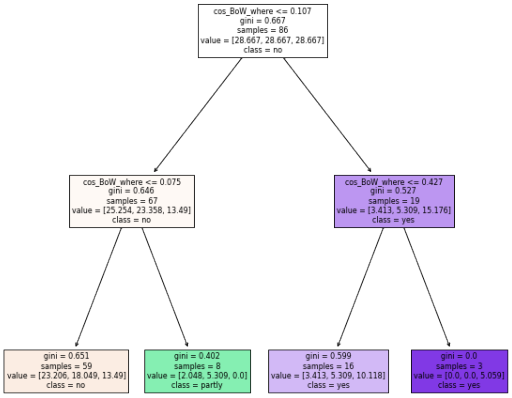
A. General Addenda



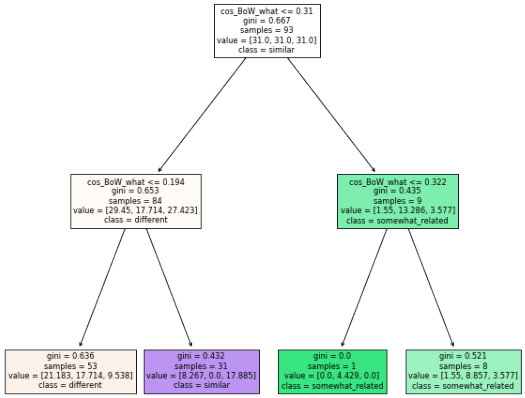
(a) Who



(b) When

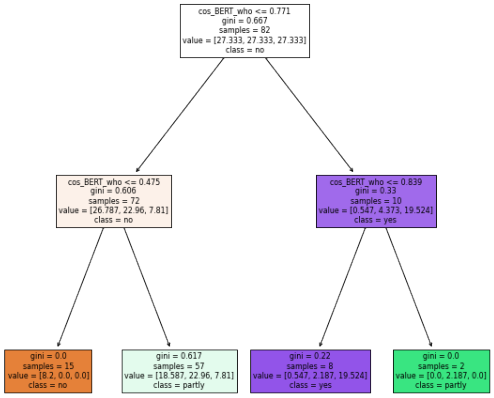


(c) Where

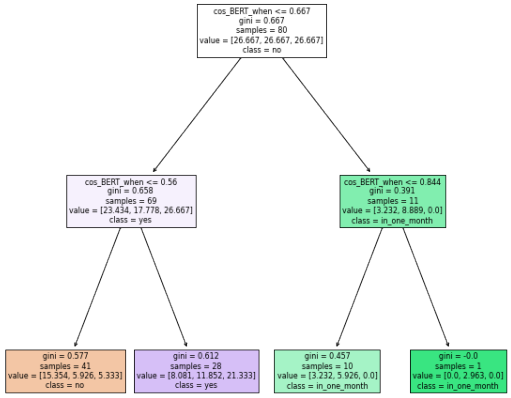


(d) What

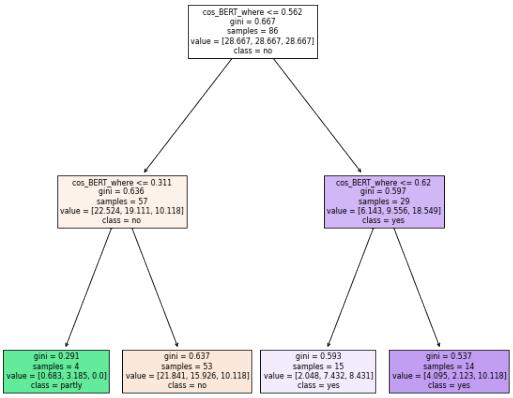
Figure A.1.: Bag-of-Words, Thresholds, Random Forest



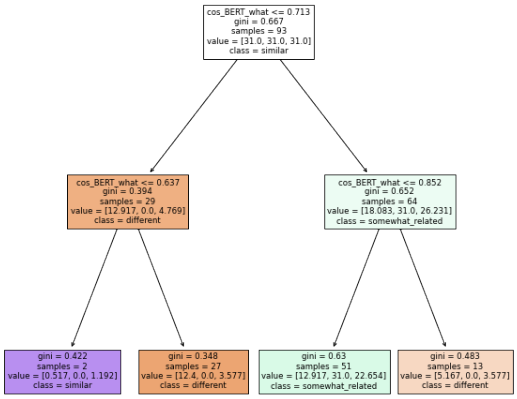
(a) Who



(b) When

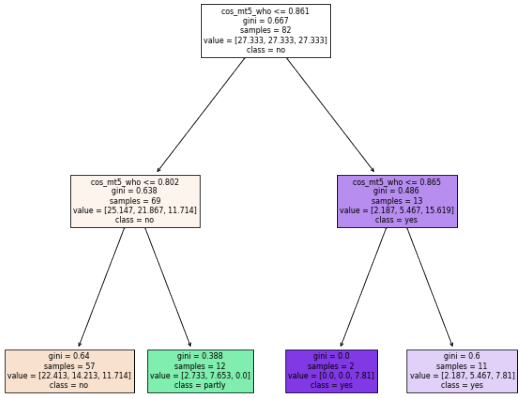


(c) Where

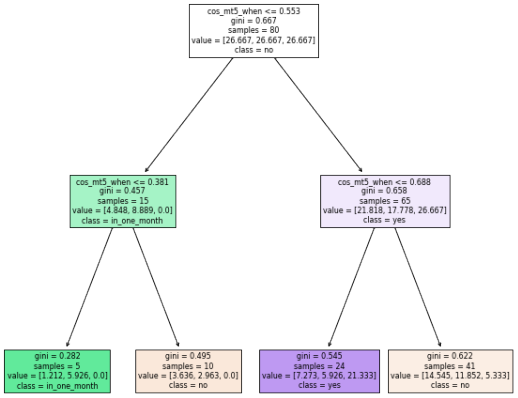


(d) What

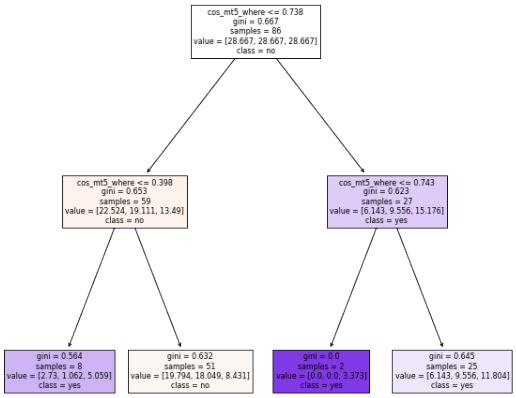
Figure A.2.: BERT, Thresholds, Random Forest



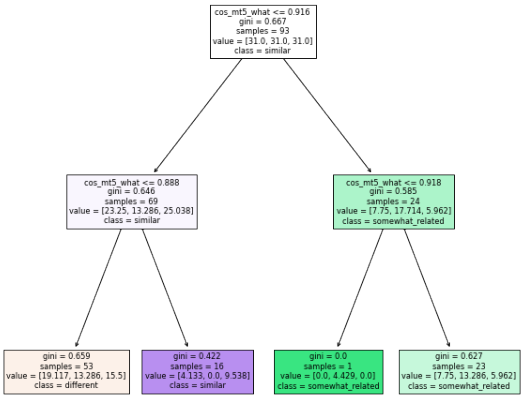
(a) Who



(b) When

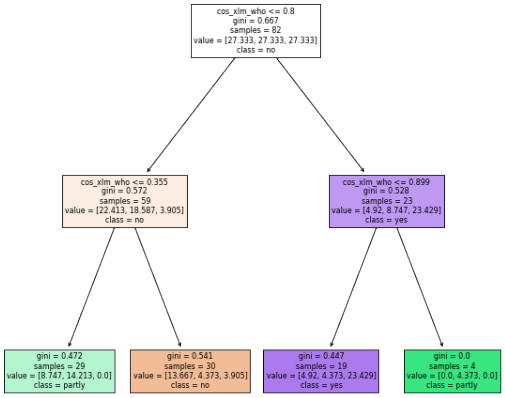


(c) Where

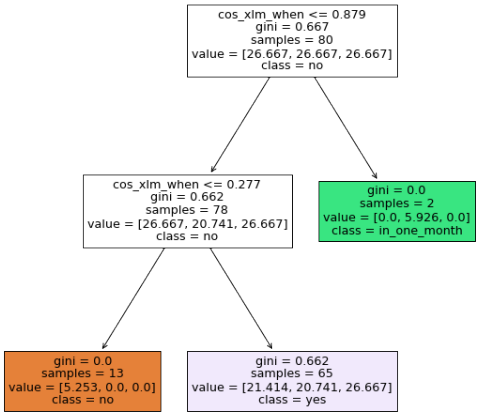


(d) What

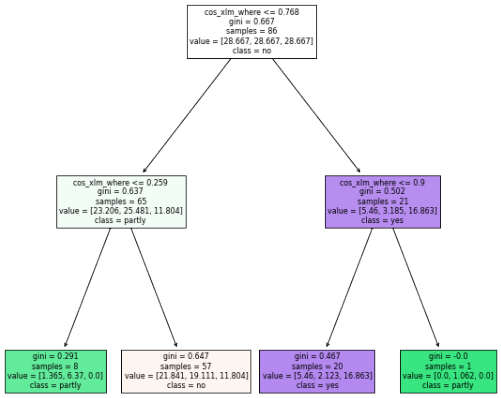
Figure A.3.: mt5-small, Thresholds, Random Forest



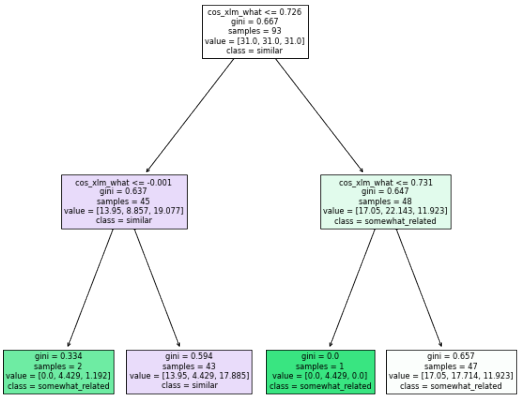
(a) Who



(b) When

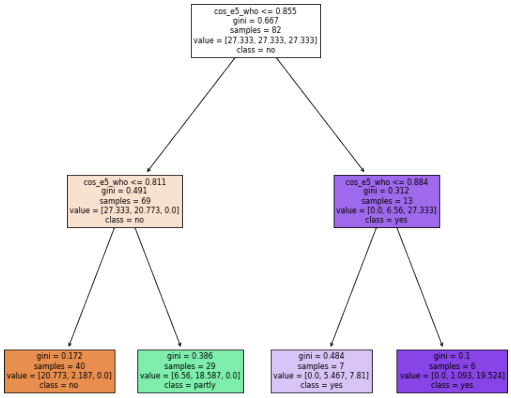


(c) Where

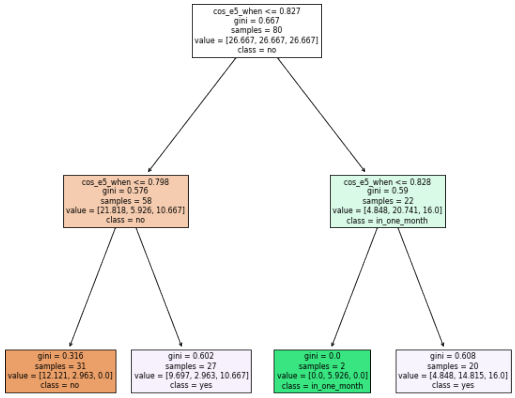


(d) What

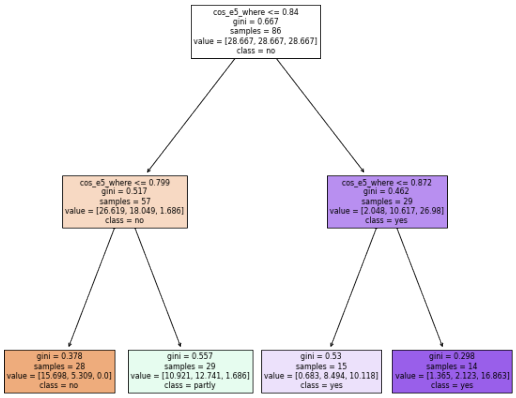
Figure A.4.: XLM-RoBERTa, Thresholds, Random Forest



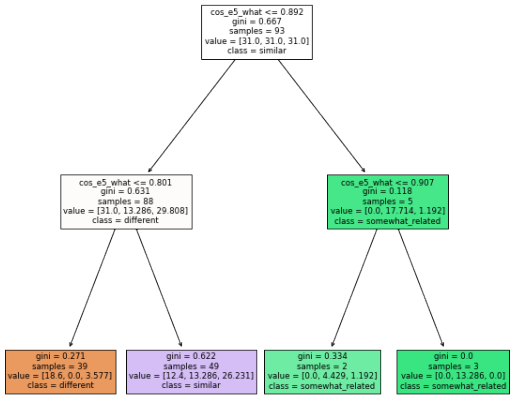
(a) Who



(b) When



(c) Where



(d) What

Figure A.5.: Multilingual e5-large, Thresholds, Random Forest

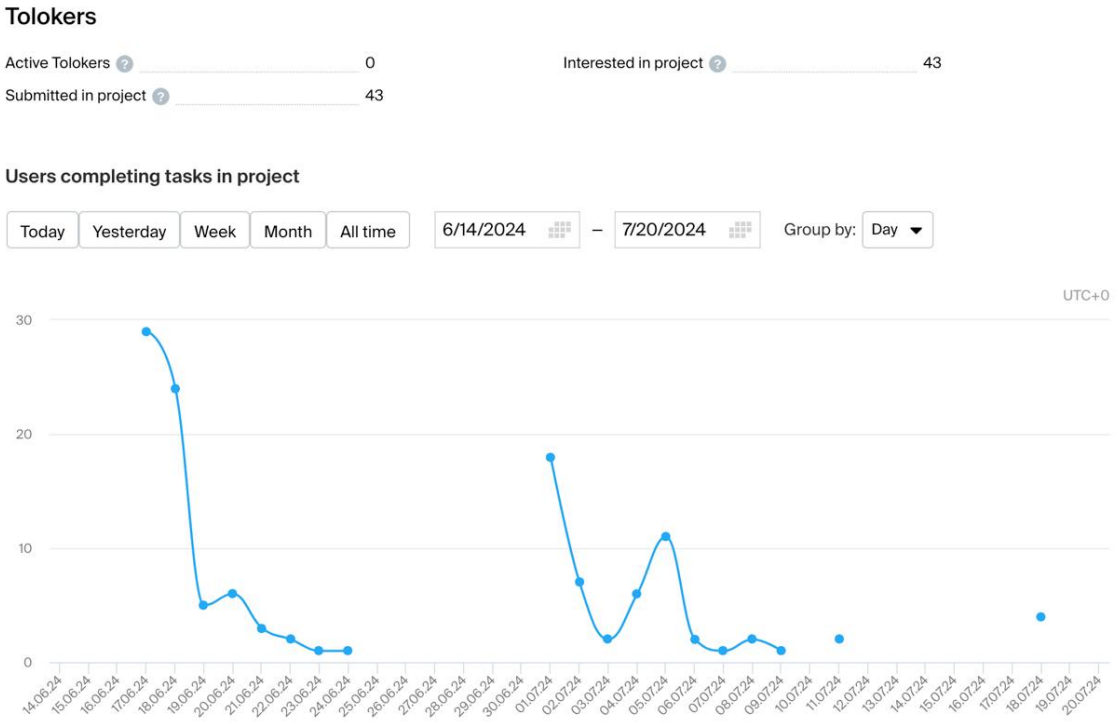


Figure A.6.: Amount of Tolokers submitting in the project per day

Figure A.7.: Task completion time per day

List of Figures

List of Tables

Bibliography

[1] X. Chen, A. Zeynali, C. Q. Camargo, F. Flöck, D. Gaffney, P. A. Grabowicz, S. A. Hale, D. Jurgens, and M. Samory. "SemEval-2022 Task 8: Multilingual News Article Similarity". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics. Online, 2022, pp. 1094–1106. DOI: 10.5281/zenodo.6507872.

[2] A. Kiulian, A. Polishko, M. Khandoga, O. Chubych, J. Connor, R. Ravishankar, and A. Shirawalmath. *From Bytes to Borsch: Fine-Tuning Gemma and Mistral for the Ukrainian Language Representation*. 2024. arXiv: 2404.09138 [cs.CL]. URL: https://arxiv.org/abs/2404.09138.

[3] S. Chen et al. *SemEval-2022 Task 8: Multilingual news article similarity*. https://zenodo.org/record/6507872. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022). 2022. DOI: 10.5281/zenodo.6507872.

[4] S. M. Mohammad and G. Hirst. *Distributional Measures of Semantic Distance: A Survey*. 2012. arXiv: 1203.1858 [cs.CL]. URL: https://arxiv.org/abs/1203.1858.

[5] J. Wang and Y. Dong. "Measurement of Text Similarity: A Survey". In: *Information* 11.9 (2020). ISSN: 2078-2489. DOI: 10.3390/info11090421. URL: https://www.mdpi.com/2078-2489/11/9/421.

[6] D. Chandrasekaran and V. Mago. "Evolution of Semantic Similarity—A Survey". In: *ACM Computing Surveys* 54.2 (2021), p. 37. DOI: 10.1145/3440755.

[7] Z. S. Harris. "Distributional Structure". In: *WORD* 10.2-3 (1954), pp. 146–162. DOI: 10.1080/00437956.1954.11659520. eprint: https://doi.org/10.1080/00437956.1954.11659520. URL: https://doi.org/10.1080/00437956.1954.11659520.

[8] K. S. Jones. "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". In: *Journal of Documentation* 28.1 (1972), pp. 11–21.

[9] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. "Okapi at TREC-3". In: *Proceedings of the Third Text REtrieval Conference (TREC-3)*. NIST. 1994, pp. 109–126.

[10] D. Sánchez, M. Batet, D. Isern, and A. Valls. "Ontology-based semantic similarity: A new feature-based approach". In: *Expert Systems with Applications* 39.9 (2012), pp. 7718–7728. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.01.082. URL: https://www.sciencedirect.com/science/article/pii/S0957417412000954.

[11] G. A. Miller. "WordNet: a lexical database for English". In: *Commun. ACM* 38.11 (1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748.

[12] R. Mihalcea and A. Csomai. "Wikify! linking documents to encyclopedic knowledge". In: CIKM '07. Lisbon, Portugal: Association for Computing Machinery, 2007, pp. 233–242. ISBN: 9781595938039. DOI: 10.1145/1321440.1321475. URL: https://doi.org/10.1145/1321440.1321475.

[13] R. Navigli and S. P. Ponzetto. "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network". In: *Artificial Intelligence* 193 (2012), pp. 217–250. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2012.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0004370212000793.

[14] Z. Wu and M. Palmer. "Verb Semantics and Lexical Selection". In: *32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico, USA: Association for Computational Linguistics, June 1994, pp. 133–138. DOI: 10.3115/981732.981751. URL: https://aclanthology.org/P94-1019.

[15] D. Sánchez and M. Batet. "A semantic similarity method based on information content exploiting multiple ontologies". In: *Expert Systems with Applications* 40.4 (2013), pp. 1393–1399. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.08.049. URL: https://www.sciencedirect.com/science/article/pii/S095741741201010X.

[16] J. Gorman and J. R. Curran. "Scaling Distributional Similarity to Large Corpora". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Ed. by N. Calzolari, C. Cardie, and P. Isabelle. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 361–368. DOI: 10.3115/1220175.1220221. URL: https://aclanthology.org/P06-1046.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL]. URL: https://arxiv.org/abs/1301.3781.

[18] Q. V. Le and T. Mikolov. *Distributed Representations of Sentences and Documents*. 2014. arXiv: 1405.4053 [cs.CL]. URL: https://arxiv.org/abs/1405.4053.

[19] J. Pennington, R. Socher, and C. Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by A. Moschitti, B. Pang, and W. Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162.

[20] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. *Enriching Word Vectors with Subword Information*. 2017. arXiv: 1607.04606 [cs.CL]. URL: https://arxiv.org/abs/1607.04606.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[23] N. Reimers and I. Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL]. URL: https://arxiv.org/abs/1908.10084.

[24] T. K. Landauer and S. T. Dumais. "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge." In: *Psychological Review* 104 (1997), pp. 211–240. URL: https://api.semanticscholar.org/CorpusID:1144461.

[25] K. Lund and C. Burgess. "Producing high-dimensional semantic space from lexical co-occurence". In: *Behavior Research Methods Instruments & Computers* 28 (June 1996), pp. 203–208. DOI: 10.3758/BF03204766.

[26] E. Gabrilovich and S. Markovitch. "Computing semantic relatedness using Wikipedia-based explicit semantic analysis". In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. IJCAI'07. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, pp. 1606–1611.

[27] M. A. Sultan, S. Bethard, and T. Sumner. "DLS@CU: Sentence Similarity from Word Alignment and Semantic Vector Composition". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Ed. by P. Nakov, T. Zesch, D. Cer, and D. Jurgens. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 148–153. DOI: 10.18653/v1/S15-2027. URL: https://aclanthology.org/S15-2027.

[28] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende. "Knowledge-enhanced document embeddings for text classification". In: *Knowledge-Based Systems* 163 (2019), pp. 955–971. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2018.10.026. URL: https://www.sciencedirect.com/science/article/pii/S0950705118305124.

[29] R. L. Cilibrasi and P. M. Vitanyi. "The Google Similarity Distance". In: *IEEE Transactions on Knowledge and Data Engineering* 19.3 (2007), pp. 370–383. DOI: 10.1109/TKDE.2007.48.

[30] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. "A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Ed. by M. Ostendorf, M. Collins, S. Narayanan, D. W. Oard, and L. Vanderwende. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 19–27. URL: https://aclanthology.org/N09-1003.

[31] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[32] Y. Le, Z.-J. Wang, Z. Quan, J. He, and B. Yao. "ACV-tree: A New Method for Sentence Similarity Modeling". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 4137–4143. DOI: 10.24963/ijcai.2018/575. URL: https://doi.org/10.24963/ijcai.2018/575.

[33] H. He and J. Lin. "Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by K. Knight, A. Nenkova, and O. Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 937–948. DOI: 10.18653/v1/N16-1108. URL: https://aclanthology.org/N16-1108.

[34] N. H. Tien, N. M. Le, Y. Tomohiro, and I. Tatsuya. "Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity". In: *Information Processing & Management* 56.6 (2019), p. 102090. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2019.102090. URL: https://www.sciencedirect.com/science/article/pii/S0306457319301335.

[35] K. S. Tai, R. Socher, and C. D. Manning. *Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks*. 2015. arXiv: 1503.00075 [cs.CL]. URL: https://arxiv.org/abs/1503.00075.

[36] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2020. arXiv: 1906.08237 [cs.CL]. URL: https://arxiv.org/abs/1906.08237.

[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG]. URL: https://arxiv.org/abs/1910.10683.

[39] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

[40] J. Camacho-Collados and M. T. Pilehvar. *From Word to Sense Embeddings: A Survey on Vector Representations of Meaning*. 2018. arXiv: 1805.04032 [cs.CL]. URL: https://arxiv.org/abs/1805.04032.

[41] P. Singh. "Systematic review of data-centric approaches in artificial intelligence and machine learning". In: *Data Science and Management* 6.3 (2023), pp. 144–157. ISSN: 2666-7649. DOI: https://doi.org/10.1016/j.dsm.2023.06.001. URL: https://www.sciencedirect.com/science/article/pii/S2666764923000279.

[42] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu. *Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond*. 2023. arXiv: 2304.13712 [cs.CL]. URL: https://arxiv.org/abs/2304.13712.

[43] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/s17-2001. URL: http://dx.doi.org/10.18653/v1/S17-2001.

[44] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. 2019. arXiv: 1804.07461 [cs.CL]. URL: https://arxiv.org/abs/1804.07461.

[45] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. *MTEB: Massive Text Embedding Benchmark*. 2023. arXiv: 2210.07316 [cs.CL]. URL: https://arxiv.org/abs/2210.07316.

[46] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. *A Survey of Large Language Models*. 2023. arXiv: 2303.18223 [cs.CL]. URL: https://arxiv.org/abs/2303.18223.

[47] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[48] OpenAI, J. Achiam, S. Adler, et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[49] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL]. URL: https://arxiv.org/abs/2310.06825.

[50] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[51] A. Üstün, V. Aryabumi, Z.-X. Yong, W.-Y. Ko, D. D'souza, G. Onilude, N. Bhandari, S. Singh, H.-L. Ooi, A. Kayid, F. Vargus, P. Blunsom, S. Longpre, N. Muennighoff, M. Fadaee, J. Kreutzer, and S. Hooker. *Aya Model: An Instruction Finetuned Open-Access Multilingual Language Model*. 2024. arXiv: 2402.07827 [cs.CL]. URL: https://arxiv.org/abs/2402.07827.

[52] M. Enis and M. Hopkins. *From LLM to NMT: Advancing Low-Resource Machine Translation with Claude*. 2024. arXiv: 2404.13813 [cs.CL]. URL: https://arxiv.org/abs/2404.13813.

[53] G. Team, R. Anil, S. Borgeaud, et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[54] I. Keraghel, S. Morbieu, and M. Nadif. "Beyond Words: A Comparative Analysis of LLM Embeddings for Effective Clustering". In: Apr. 2024, pp. 205–216. ISBN: 978-3-031-58546-3. DOI: 10.1007/978-3-031-58547-0_17.

[55] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. *Emergent Abilities of Large Language Models*. 2022. arXiv: 2206.07682 [cs.CL]. URL: https://arxiv.org/abs/2206.07682.

[56] Common Crawl. *Common Crawl Dataset*. Available at: https://commoncrawl.org [Accessed: August 24, 2024]. 2023.

[57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. "Language Models are Unsupervised Multitask Learners". In: 2019. URL: https://api.semanticscholar.org/CorpusID:160025533.

[58] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*. 2015. arXiv: 1506.06724 [cs.CV]. URL: https://arxiv.org/abs/1506.06724.

[59] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. 2020. arXiv: 2101.00027 [cs.CL]. URL: https://arxiv.org/abs/2101.00027.

[60] Wikipedia contributors. *Wikipedia, The Free Encyclopedia*. https://www.wikipedia.org. Accessed: August 24, 2024. 2024.

[61] J. Gatto, O. Sharif, P. Seegmiller, P. Bohlman, and S. M. Preum. "Text Encoders Lack Knowledge: Leveraging Generative LLMs for Domain-Specific Semantic Textual Similarity". In: *arXiv preprint arXiv:2309.06541* (2023).

[62] T. Gao, X. Yao, and D. Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6894–6910. DOI: 10.18653/v1/2021.emnlp-main.552. URL: https://aclanthology.org/2021.emnlp-main.552.

[63] W. G. Bleyer. *Newspaper Writing and Editing*. Boston, MA: Houghton Mifflin Company, 1913.

[64] M. Capelle, F. Frasincar, M. Moerland, and F. Hogenboom. "Semantics-based news recommendation". In: *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*. WIMS '12. Craiova, Romania: Association for Computing Machinery, 2012. ISBN: 9781450309158. DOI: 10.1145/2254129.2254163. URL: https://doi.org/10.1145/2254129.2254163.

[65] M. U. Sen, H. Y. Erdinc, B. Yavuzalp, and M. C. Ganiz. "Combining Lexical and Semantic Similarity Methods for News Article Matching". In: *Data Science – Analytics and Applications*. Ed. by P. Haber, T. Lampoltshammer, and M. Mayr. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 29–35. ISBN: 978-3-658-27495-5.

[66] M. AL-Smadi, Z. Jaradat, M. AL-Ayyoub, and Y. Jararweh. "Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features". In: *Information Processing & Management* 53.3 (2017), pp. 640–652. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2017.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0306457316302382.

[67] R. Singh and S. K. Singh. "Text Similarity Measures in News Articles by Vector Space Model Using NLP". In: *Journal of The Institution of Engineers (India): Series B* 102.2 (2021), pp. 329–338. DOI: 10.1007/s40031-020-00501-5. URL: https://doi.org/10.1007/s40031-020-00501-5.

[68] O. Darwish, Y. Tashtoush, M. Maabreh, R. Al-essa, R. Aln'uman, A. Alqublan, M. Abualkibash, and M. Elkhodr. "Identifying Fake News in the Russian-Ukrainian Conflict Using Machine Learning". In: *Advanced Information Networking and Applications*. Ed. by L. Barolli. Cham: Springer International Publishing, 2023, pp. 546–557.

[69] J. Sido, M. Seják, O. Pražák, M. Konopík, and V. Moravec. *Czech News Dataset for Semantic Textual Similarity*. 2022. arXiv: 2108.08708 [cs.CL]. URL: https://arxiv.org/abs/2108.08708.

[70] H. Roberts, R. Bhargava, L. Valiukas, D. Jen, M. M. Malik, C. Bishop, E. Ndulue, A. Dave, J. Clark, B. Etling, R. Faris, A. Shah, J. Rubinovitz, A. Hope, C. D'Ignazio, F. Bermejo, Y. Benkler, and E. Zuckerman. *Media Cloud: Massive Open Source Collection of Global News on the Open Web*. 2021. arXiv: 2104.03702 [cs.SI]. URL: https://arxiv.org/abs/2104.03702.

[71] Y. C. Z. Xu Z. Yang and Z. Chen. "HFL at SemEval-2022 Task 8: A Linguistics-Inspired Regression Model with Data Augmentation for Multilingual News Similarity". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics, 2022, pp. 1114–1120.

[72] I. Singh, Y. Li, M. Thong, and C. Scarton. "GateNLP-UShef at SemEval-2022 Task 8: Entity-Enriched Siamese Transformer for Multilingual News Article Similarity". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics. 2022, pp. 1121–1128.

[73] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. *Language-agnostic BERT Sentence Embedding*. 2022. arXiv: 2007.01852 [cs.CL]. URL: https://arxiv.org/abs/2007.01852.

[74] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, and T.-Y. Liu. *R-Drop: Regularized Dropout for Neural Networks*. 2021. arXiv: 2106.14448 [cs.LG]. URL: https://arxiv.org/abs/2106.14448.

[75] M. Kuimov, D. Dementieva, and A. Panchenko. "SkoltechNLP at SemEval-2022 Task 8: Multilingual News Article Similarity via Exploration of News Texts to Vector Representations". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics. 2022, pp. 1136–1144.

[76] Z. Chen, W. Chen, Y. Sun, H. Xu, S. Zhou, B. Chen, C. Sun, and Y. Liu. "ITNLP2022 at SemEval-2022 Task 8: Pre-trained Model with Data Augmentation and Voting for Multilingual News Similarity". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics. 2022, pp. 1184–1189.

[77] M. Di Giovanni, T. Tasca, and M. Brambilla. "DataScience-Polimi at SemEval-2022 Task 8: Stacking Language Models to Predict News Article Similarity". In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics. 2022, pp. 1229–1234.

[78] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou. Online: Association for Computational Linguistics, June 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41. URL: https://aclanthology.org/2021.naacl-main.41.

[79] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: 1911.02116 [cs.CL]. URL: https://arxiv.org/abs/1911.02116.

[80] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei. *Multilingual E5 Text Embeddings: A Technical Report*. 2024. arXiv: 2402.05672 [cs.CL]. URL: https://arxiv.org/abs/2402.05672.

[81] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy. *LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders*. 2024. arXiv: 2404.05961 [cs.CL]. URL: https://arxiv.org/abs/2404.05961.

[82] V. Aryabumi, J. Dang, D. Talupuru, S. Dash, D. Cairuz, H. Lin, B. Venkitesh, M. Smith, K. Marchisio, S. Ruder, A. Locatelli, J. Kreutzer, N. Frosst, P. Blunsom, M. Fadaee, A. Üstün, and S. Hooker. *Aya 23: Open Weight Releases to Further Multilingual Progress*. 2024. arXiv: 2405.15032 [cs.CL].