Assessing the Difficulty of Spans Within Nested Text Markup Data Models

A Preprint

Andrey M. Tabachenkov
Department of Mathematical Methods of Forecasting
Moscow State University,
Machine Learning and Semantic Analysis
MSU Institute for Artificial Intelligence
a.tabachenkov@iai.msu.ru

Archil I. Maysuradze
Department of Mathematical Methods of Forecasting
Moscow State University,
Machine Learning and Semantic Analysis
MSU Institute for Artificial Intelligence
useraim@mail.ru

Abstract

While improving the performance of machine learning (ML) models, researchers assess the difficulty of objects in a dataset. There is no universally agreed-upon definition of what constitutes a difficult object. The most commonly used term is "difficulty"; although other terms such as "hardness" and "challenging cases" are also used. There are numerous methods for assessing object difficulty, ranging from traditional approaches to more recent techniques based on model training. These methods are applied in a variety of domains and to various data structures, helping to improve solutions by filtering out or re-labeling difficult objects. This paper extends these methods to the humanities, where researchers encounter inherently complex data schemas. Therefore, the use of conventional methods is complicated by the need to transform complex data into vector representations. In particular, we consider labeled text spans, for which the vector representation must take the entire context into account. To address this challenge, we study the use of language models to create vector representations of textual fragments that consider the context within the overall difficulty assessment pipeline. This yields a more accurate representation of a text's meaning and complexity, enabling a more precise estimation of the difficulty of each data instance.

Keywords Evaluation of difficulty for individual data instances · Text labeling · Digital humanities · Machine Learning · Natural language processing

1 Introduction

The evaluation of object complexity constitutes a critical component of data mining, as it provides insights into the performance of trained models. Identifying difficult objects is particularly valuable for the re-annotation of data and subsequent development of models. The methods employed for this assessment have evolved to encompass a broad spectrum of tasks, domains, and entities exhibiting varied characteristics.

In the realm of machine learning and data analysis, various methodologies exist for evaluating the difficulty associated with individual data instances. These methodologies range from traditional statistical approaches to contemporary techniques utilising neural networks. The evaluation process is further complicated by the

absence of a universally accepted definition of a "difficult object", which has led to diverse interpretations among researchers.

The terminology surrounding this concept remains contentious, with some scholars referring to such objects as "challenging" or "atypical", while others prefer the term "difficult". Nonetheless, "difficulty" appears to be the most widely endorsed designation within the literature.

The application of these machine learning techniques within the humanities is complicated by the need for precise knowledge formalisation. For instance, nested data are characterized by a hierarchical or multilevel structure, that is, are organized at more than one level. We consider following levels: the first and the second sub-levels of the level of spans and the level of elements, which are described in subsection 2.2.

In this article, we focus on the span level of a chain of nested data models, considering the text spans along with their corresponding tags as entities for difficulty assessment. The application of traditional methods to evaluate the difficulty of objects within a chain of nested data models encounters several challenges, including the phenomenon of multi-assessment and the necessity of data vectorization. This article will explore strategies to surmount these obstacles.

Firstly, we propose integrating assessor consistency into the loss function 2.3, thereby ensuring that the model is trained to yield reliable results across various assessors. Secondly, we advocate for the utilisation of language models to generate vector representations of text spans 2.4 and their contexts. This approach will aid in capturing the semantic essence of the text, facilitating easier comparisons between distinct objects.

The authors of considered articles about difficulty demonstrate varying interpretations of the concept of object difficulty. In this paper the difficulty of an object refers to the complexity encountered by a machine learning model when processing that object in a specific task. We focus exclusively on model-specific and task-specific methods since we consider text's span with single tag as object the current task is the span classification. Besides that, model specificity entails the need for training span classifier.

2 Related works

2.1 Difficulty assessement methods

Methods for assessing the difficulty have been addressed in a limited number of publications, such as the work Seedat et al. [2024]. We have drawn upon the mathematical and methodological foundations underlying these methods:

- Inclusion of distribution support estimation
- Inclusion of distribution density estimation for the object (or its features)
- Utilisation of reconstruction error as a measure of object difficulty
- Task-agnosticism (It is important to note that, in most instances, the task-agnostic nature of a method is directly related to the absence of a labelled or target feature for the object)
- Model-agnosticism
- Computation of statistics for a trained model (with respect to its layers; we focus here on model-specific approaches)
- Generation of (pseudo)difficult objects (for the purposes of training or validation).

The overview is presented in the Tables 1 and 2, respectively, for the articles under consideration and for the methods from scikit-learn. Additionally, we have also systematized the margin-based approach that is task- and model-specific.

All the methods discussed rely on vectorised objects as input, along with their accompanying labels.

Furthermore, when employing these methods directly, the assessment of difficulty is typically conducted on the entire dataset without exception, with the notable exception of the article Lee et al. [2018], which utilised a validation dataset containing previously known complex objects to refine hyperparameters (utilising objects from alternative datasets). In instances where pseudo-difficult objects (or pseudo-outliers) were employed, these were generated automatically using noise or an algorithm akin to that described in article Zhou and Wang [2024]. In other scenarios, the authors resorted to unsupervised learning methods or abstained from any training altogether, merely calculating statistics derived from pre-trained models.

Таблица 1: Considered articles and their correspondences to characteristics

Article	Support estimation	Distribution estimation	Recon-struction	Task-agnostic	Model-agnostic	Statistics	Generation
Support vector data description Tax and Duin [2004]	✓			✓	✓		
Task-agnostic out-of-distribution detection using kernel density estimation Erdil et al. [2021]		✓		✓		✓	
Robust, deep and inductive anomaly detection Chalapathy et al. [2017]			✓	✓	✓		
A simple unified framework for detecting out-of-distribution samples and adversarial attacks Lee et al. [2018]		✓				✓	
Deep semi-supervised anomaly detection Ruff et al. [2019]	✓			✓	✓		
Rapp: Novelty detection with reconstruction along projection pathway Kim et al. [2019]			✓	✓		✓	
Unsupervised anomaly detection with generative adversarial networks to guide marker discovery Schlegl et al. [2017]				✓	✓		
Estimating example difficulty using variance of gradients Agarwal et al. [2022]						✓	
Grod: Enhancing generalization of transformer with out-of-distribution detection Zhou and Wang [2024]	✓					✓	✓

Таблица 2: Considered functions of sklearn and their correspondences to characteristics

Function	Support estimation	Distribution estimation	Recon-struction	Task-agnostic	Model-agnostic	Statistics	Generation
One Class SVM	✓			✓	✓		
Elliptic envelope		✓		✓	✓		
Isolation forest				✓	✓		
Local outlier factor				✓	✓		

In contrast, when evaluating the efficacy of these algorithms, the authors relied on previously acknowledged difficult objects, either by employing alternative datasets or by designating one class as complex or noisy, subsequently excluding it during the training phase, or by using explicit annotations.

Since we consider model-specific and task-specific difficulty, this article will further examine the works Lee et al. [2018], Agarwal et al. [2022], and the margin-based approach.

2.2 Nested data models

In neuroscience synapses (level 1) are organized, or nested, in cells (level 2) Aarts et al. [2014]. The annotation of multiple spans in content analysis enhances the identification of human values in textual data Maysuradze et al. [2024], Rink et al. [2024], Vorontsov et al. [2025].

Within the framework of nested data models adopted for this study, analysis is confined to the span level and the element level, where elements comprise multiple spans. This framework was developed based on existing

datasets and comprises a sequence of text markup data models that are nested within one another, aligning with varying levels of markup complexity.

First sub-level of level of spans At this level, each document contains only a single markup, which is composed of spans. Each span is designated with only one tag, commonly referred to as a SpanTag. This structure aligns with many classical datasets characterised by simplicity. These datasets are predominantly utilised in Named Entity Recognition (NER) tasks. An example of such a dataset is the Kaggle NER Corpus Nadeau and Sekine [2009].

Second sub-level of level of spans At this level, a document may feature multiple annotations, introducing the concept of multi-assessority. Each span can be assigned zero or more tags; the absence of tags indicates that all spans within the dataset share the same tag, which is therefore omitted. Conversely, the presence of multiple tags corresponds to scenarios of multi-label classification or a complex hierarchical structure within the tagging system. For example, the RuSentNE dataset Golubev et al. [2023] assigns both Named Entity Recognition (NER) tags and sentiment tags to each span.

Level of elements At this level, each annotation comprises elements rather than spans, with an element being defined as a set of spans. The interpretation of elements may vary across different datasets. Each element is also assigned tags (ElementTag). Typical instances of elements include coreference clusters, which pertain to the coreference resolution task, as well as relations, frames, and multi-fragments Maysuradze et al. [2024]. Examples of datasets featuring elements include the Ruethics and RWSD tasks from the MERA project Fenogenova et al. [2024], as well as the ADE dataset Gurulingappa et al. [2012], NEREL Loukachevitch et al. [2023], RURED Gordeev et al. [2020], the RWSD task from RuSuperGlue, SCIERC Luan et al. [2018], and SemEval 2010 task 8 Hendrickx et al. [2019]. These datasets not only include individual spans but also incorporate the relationships between them and/or coreference clusters.

2.3 Multi-assessorship

In practice, it is not uncommon for situations to arise where the texts within a dataset are identical, yet their markup may vary. This phenomenon can be described as multi-assessorship without explicit information regarding the assessors, which corresponds to the second sub-level of the span level within a chain. Although this aspect is typically overlooked during the training of models and the assessment of object difficulty, we propose an alternative approach. Specifically, we recommend introducing an additional stage of multi-assessorship processing prior to the vectorisation of spans and the application of methods for assessing difficulty.

Firstly, we advocate for the utilisation of multiple markups of individual objects through the implementation of specialised loss functions during training, alongside various forms of consistency and consensus. An example of such an approach is articulated in article Le et al. [2023].

In that article, the authors examined the task of segmentation using bounding boxes and explored the classification of these boxes. For each image, they organised similar bounding boxes—derived from different markups of the images—into clusters based on the Intersection over Union (IoU) score:

$$IoU((begin_{x,1}, end_{x,1}, begin_{y,1}, end_{y,1}), (begin_{x,2}, end_{x,2}, begin_{y,2}, end_{y,2}) = \frac{IntersectionArea}{Area_1 + Area_2 - IntersectionArea}$$

$$IntersectionArea = (min(end_{x,1}, end_{x,2}) - max(begin_{x,1}, begin_{x,2}))(min(end_{y,1}, end_{y,2}) - max(begin_{y,1}, begin_{y,2}))$$

$$Area_1 = (end_{x,1} - begin_{x,1})(end_{y,1} - begin_{y,1})$$

$$Area_2 = (end_{x,2} - begin_{x,2})(end_{y,2} - begin_{y,2})$$

Subsequently, for each cluster, the authors computed an averaged bounding box, taking into account the reliability weights assigned to the assessors. For the computed bounding box and each class $k \in \{1, \ldots, K\}$, the authors calculated a confidence score denoted by:

$$c_k = c_0 \min(T, N),$$

where T represents the number of experts corresponding to the cluster in question, who labelled the boxes within that cluster with class k. This coefficient was then utilised as a weight in the loss function for the

averaged bounding box. If T is sufficiently large and numerous assessors select very similar bounding boxes with the same label k, the resulting averaged bounding box assigned label k will exhibit a high confidence score.

According to Le et al. [2023], the incorporation of confidence scores into the loss function enhances the robustness and generalisation capability of the model.

2.4 Span vectorisation

There exists a variety of methods for deriving vector representations of spans through the use of language models. In the majority of instances involving spans, aggregation functions are utilised (which are used to aggregate vector representations of tokens in spans). Examples of such methodologies are detailed in the following studies: Joshi et al. [2020] (which discusses the application of boundary tokens and positional embeddings for spans' lengths), Eberts and Ulges [2020] (which explores max pooling along with positional embeddings of spans' lengths), and Toshniwal et al. [2020] (where the authors examined six distinct methods: average pooling, attention pooling, max pooling, endpoint, diff-sum, and coherent). We note, that in different tasks such methods may have different levels of success.

2.5 Margin-based approach

We further propose a methodological approach that, although classical in nature, has received scant attention in the extant research. Let us consider a model trained on the multi-class classification task: $a(x) = \arg\max_{y \in Y} g_y(x)$, where $g_y(\cdot)$ is a discriminant function corresponding to the class $y \in Y$ (for example, a set of such discriminant functions can be a pre-softmax layer in a neural network). Let $\hat{y}$ be a true class of object x. Then we introduce $M_y(x) = g_{\hat{y}}(x) - g_y(z)$ - margin of object x by class y. Then the (total) margin of object x is $M(x) = \min_{y \neq \hat{y}} M_y(x)$.

The margin is deemed to be positive if and only if the object is accurately classified. The absolute value of the margin can be interpreted as an indicator of the model's confidence in its prediction.

Consequently, when the margin is significantly less than zero, the object is regarded as an outlier in relation to the model (indicating that the model has made an error, albeit with a degree of certainty). A small absolute value of the margin may suggest that the model is not fully trained or that the object poses a challenge to the model, preventing it from making a confident decision.

3 Problem statement

Let we have text $T = \{\tau_1, ..., \tau_l\}$, where τ_i is a token (tokenization corresponds to the considered tokenizer) and its text span $s = \{\tau_{begin}, ..., \tau_{end}\}$ with the SpanTag $y \in Y$. Thus, we consider triplet $(T, (begin, end), y)$.

It is required to estimate the difficulty of these spans with tags $DiffScore : X \to \mathbb{R}$, where X is the set of spans with tags.

If $D_{non-diff}$ is the set of nominal non-difficult objects and D_{diff} is the set of nominal difficult objects then the quality criterion is a ROC-AUC Fawcett [2006] applied to difficulty scores of objects, considering objects from $D_{non-diff}$ as negative and objects from D_{diff} as positive 2.1.

Let $ConfScore : X \to [0, 1]$ be the confidence score of pair span-tag 2.3. Then we also check hypothesises that $ConfScore$ correlates with $DiffScore$ and that objects with lower confidence scores have higher difficulty scores.

4 Adaptation of methods

4.1 Vectorization

In this article, we focus exclusively on spans, which correspond to the span level within a chain. It has been previously noted that most methods for assessing difficulties operate using vector representations of objects. Consequently, we employed Large Language Model (LLM) frameworks to vectorise texts, tokens, and spans 2.4.

Let us have text $T = \{\tau_1, ..., \tau_l\}$, where τ_i is a token (tokenization corresponds to the LLM's tokenizer). Let M be the transformer-based language model that vectorizes considered texts: $M(\{\tau_1, ..., \tau_l\}) = \{e_1, ..., e_l\}$, where e_i is the embedding of i-th token from the space of meanings E. Let us consider span $s = \{\tau_{begin}, ..., \tau_{end}\} \subset T$. It's proposed to extract vector representation of span s using aggregation of embeddings of corresponding tokens: $A(\{e_{begin}, ..., e_{end}\}) = e^s$, where e^s is the embedding of span s, A is the aggregation function.

The vector representations of spans can subsequently be submitted to the classifier. In the current dataset, where each span is associated with only one SpanTag (at the level of spans from the chain), the classifier is trained to perform a multi-class classification task. Given the available SpanTags $tag_1, ..., tag_b$, the classifier returns a probability distribution over the corresponding tags: $(p_1, ..., p_b, p_o)$, where $p_i \geq 0; p_o \geq 0; p_o + \sum_{i=1}^{b} p_i = 1$ and p_i denote the probability that span s is labeled with tag_i and p_o represents the probability that the span is not labeled with any of the tags tag_i". The final probability corresponds to the special tag "Other". Although this tag is absent from most datasets, it has been incorporated here to explicitly indicate instances where a span demarcated by the specified boundaries is not represented in the document's markups.

Subsequently, methods for assessing difficulty can be applied to the vector representations of spans and the classifier mentioned above.

4.2 Multi-assessorship

To adapt difficulty assessment methods to multi-assessor document markups, it is proposed to use an approach similar to that described in section 2.3 aggregating similar spans and calculating their confidence (or consistency) scores which can be used as weights of spans in cross-entropy loss used to train span classifier.

Moreover, certain peculiarities emerge when attempting to measure the confidence scores of "span-tag"pairs, which we also address in this discussion. To begin with, in the aforementioned article, the authors utilised scores related to assessors' reliabilities. However, as previously noted, lower levels of multi-assessorship, including the second sub-level at the span level, are characterised by a lack of known assessors. We propose addressing this issue by assuming uniform reliability scores for all anonymous assessors responsible for document markups.

Additionally, an arbitrary document may have a varying number of markups. To accommodate this, we adaptively adjust the parameters N and c_0 (as defined in the previous formula) in accordance with the increasing number of markups within a document, potentially considering the ratio of T to the total number of markups pertinent to the current document.

Finally, a pertinent question arises regarding which objects' difficulty within the chain we should assess and the methodology employed in this assessment. The Intersection over Union (IoU) score and average aggregation methods may also be applicable when processing fragments analogous to those associated with the previously considered bounding boxes.

In summary, we propose an adaptation to multi-assessorship as follows: firstly, we will define clusters of spans by utilising the IoU score as a measure of span similarity. The IoU score is computed based on the boundaries of spans:

$$IoU(\{\tau_{begin_1}, ..., \tau_{end_1}\}, \{\tau_{begin_2}, ..., \tau_{end_2}\}) = \begin{cases} 0, begin_1 \geq end_2 \\ 0, begin_2 \geq end_1 \\ \frac{min(end_1, end_2) - min(begin_1, begin_2)}{max(end_1, end_2) - max(begin_1, begin_2)}, other \end{cases}$$

Next, for each cluster, we calculate the consensus span by averaging the beginning and ending indices. In addition, we compute a confidence score for each pair of cluster and tag, where the tag is present within the spans of the clusters: $ConfScore(C_{tag}) = \frac{|C_{tag}|}{|Markups_{doc}|}$, where C_{tag} denotes the set of spans within a cluster that have a specific tag, and $Markups_{doc}$ represents the set of all markups within the document. We have taken into account that within a single cluster, all spans originate from different markups, as spans within a single markup cannot intersect in the datasets under consideration.

Further in the text, objects with high confidence scores (greater than 0.7) will be called consistent, objects will low confidence scores (less than 0.7) will be called inconsistent. Besides that, we introduce an uncertainty score $UncerScore(\cdot) = 1 - ConfScore(\cdot)$.

5 Experiments

This study is also extended to an evaluation of the performance of various difficulty assessment methods using real-world data from our collection. The following outlines the procedures undertaken for data preprocessing, language model training, validation of the difficulty assessment methods, hypothesis checking, and the interpretability of the difficulty scores:

Data Preprocessing:

- The datasets were transformed into a span-level data model, moving away from the chain-based structure.

Language Model Training and Validation:

- A single dataset was selected to serve as the primary reference.
- The model architecture was implemented according to the methodologies outlined in the vectorisation section, employing a span embedder coupled with a classifier at the output layer.
- Both training and validation procedures were conducted on this primary dataset, integrating the precomputed confidence scores from the assessors.

Validation of Difficulty Assessment Methods:

- The efficacy of the difficulty assessment methods was evaluated using methodologies aligned with those found in the established literature.
- Non-difficult objects were identified within the primary dataset, while the remaining datasets were used to source (pseudo)difficult objects.
- Furthermore, a technique that introduces noise to fragment embeddings was employed for additional assessment.
- The performance of the various methods was compared on this corpus of (pseudo)difficult data. Specifically, four distinct methods were validated: Variance of gradients Agarwal et al. [2022], Gaussian method Lee et al. [2018], Margin-based approach, A variant of the margin-based approach utilising absolute values of margins.

Correlations between difficulty assessing methods:

- For each pair of difficulty assessing methods the rank correlation was calculated between corresponding difficulty scores of objects from validation set.

Interpretability of Difficulty Assessment Methods:

- The various difficulty assessment methods were applied to objects from the primary dataset to evaluate the interpretability of the high difficulty scores obtained.

Connection between difficulty and consistency:

- In this experiment the connection between high model-specific difficulty and low consistency of objects was explored.
- Firstly, for each difficulty assessing methods the rank correlation was calculated between difficulty scores and uncertainty scores of objects.
- Secondly, we applied ROC-AUC to difficulty scores of validation set's objects, considering consistent objects as negative ones and inconsistent objects as positive ones.
- Thirdly, the Kolmogorov-Smirnov test was used to check whether difficulty scores of consistent and inconsistent objects are from the same distribution or not.

5.1 Data preprocessing

We examined two datasets: the Kaggle NER Corpus Nadeau and Sekine [2009] and the ADE datasets Gurulingappa et al. [2012]. The Kaggle NER Corpus was selected as primary as it directly aligns with the specific challenge of span identification and classification. In contrast, the ADE datasets served for validating our methods of assessing difficulty.

Таблица 3: Difficult and noisy objects with low consistency scores

Document with span	Tag and score
Leta Hong Fincher has more.	per-0.5, org-0.5
Last month, Lebanese officials claimed victory in the fighting, but daily firefights have continued since then.	tim-0.33
Bird flu has killed at least 81 people in East Asia and Turkey since 2003.	nat-0.33
Television reports say few onlookers turned out for a glimpse of the Prince of Wales and his long-time companion, the Duchess of Cornwall, whom he married earlier this year.	org-0.5
Television reports say few onlookers turned out for a glimpse of the Prince of Wales and his long-time companion, the Duchess of Cornwall, whom he married earlier this year.	geo-0.5, per-0.5
Television reports say few onlookers turned out for a glimpse of the Prince of Wales and his long-time companion, the Duchess of Cornwall, whom he married earlier this year.	geo-0.5, per-0.5
Muslim separatists in Indian Kashmir are fighting for an independent Kashmir or its merger with neighbouring Pakistan.	org-0.5

5.1.1 Kaggle NER Corpus

This dataset Nadeau and Sekine [2009] represents the level of span annotations. Notably, spans do not overlap within a single markup due to the authors employing BIO notation. In our approach, we opted to convert this dataset to the level of the spans data model to consider text spans with tags.

The dataset includes 8 distinct tags in addition to a special "Other"tag: "art"(artifact), "eve"(event), "geo"(geographical object), "gpe"(geo-political entity), "nat"(natural phenomenon), "org"(organization), "per"(person), and "tim"(time). This clearly positions the dataset within the realm of Named Entity Recognition (NER) tasks.

For each document and its corresponding annotations, we implemented the algorithm outlined in the section discussing adaptation to multi-assessorship. Approximately 1% of the documents feature multiple annotations, indicating that the vast majority of span-tag pairs are consistent.

We defined span clusters using the Intersection over Union (IoU) score, adding a span to a cluster if its IoU score with the cluster's spans is at least 0.7. This threshold was selected to ensure that only genuinely similar fragments are grouped together in one cluster.

This section also presents examples of inconsistent span-tag pairs, which we identify as challenging or noisy in the context of multi-assessor annotations. These examples can be found in Tab. 3. In most cases of low confidence scores, a small number of assessors likely selected an incorrect tag (indicating noisy markups), or there was disagreement among assessors regarding tag selection (reflecting difficult objects). Additionally, a small number of assessors may have highlighted the span, which could signal either assessor error or the complexity of the task.

5.1.2 ADE

The dataset described in Gurulingappa et al. [2012] consists of two subsets: the first includes drug-dosage pairs with 501 documents, while the second contains drug-effect pairs comprising 10,959 documents. A streamlined data model representing this dataset is structured around the elements of a chain, treating the relations of drug-dosage and drug-effect as individual elements. However, due to the focus on spans, the dataset has been refined to the level of spans, wherein each markup is associated with all spans found within the existing element-relations.

In contrast, the documents in the ADE dataset are biomedical in nature, whereas the Kaggle NER Corpus Nadeau and Sekine [2009] consists of texts related to news and politics. In evaluating the complexity of the objects based on a model trained on the Kaggle corpus, this significant difference suggests that the biomedical spans from the ADE dataset are inherently more challenging to model. Consequently, methods designed for difficulty assessment should assign higher scores to the objects from the ADE dataset.

5.2 Language model training and validation

We used BERT model Devlin et al. [2019] and DeBERTa model He et al. [2021] as the language models $\mathcal{M}$ for their efficacy in generating token vector representations. We also considered all 6 main aggregation methods from Toshniwal et al. [2020]. Besides that, we considered SpanBERT Joshi et al. [2020] model with endpoint aggregation. A Multi-Layer Perceptron (MLP) with two linear layers and GeLU Hendrycks and Gimpel [2016] as activation was served as the classifier. To enhance the quality and generalisability of the derived span embeddings, specific layers of the language models were unfrozen during training.

The training set consisted of approximately 5,000 texts from the Kaggle NER Corpus Nadeau and Sekine [2009], with a further 2,000 texts allocated for validation. A strict separation was maintained between training and validation documents to preclude any data leakage. For both phases, the input was formatted as a triplet $(T, (begin, end), tag)$, where T denotes the text, $(begin, end)$ defines the span boundaries, and tag specifies the label—either "Other"or one of the eight predefined NER tags. It is noteworthy that while these tags do not directly influence the model's internal mechanics, they are fundamental to the computation of the loss function during classifier training and, by extension, the formation of the span vector representations.

As the Kaggle NER dataset Nadeau and Sekine [2009] does not inherently include an "Other"tag for spans, a negative sampling procedure was implemented. To generate triplets of the form $(T, (begin, begin + l), Other)$, the distribution of span lengths, denoted U_{len}, was first estimated from the training documents. A length l was sampled from U_{len}, and a starting index $begin$ was drawn from a uniform distribution $U[0, L - l]$, where L signifies the total number of tokens in text T.

During training, negative samples were regenerated anew for each epoch, producing one sample per training document. For the validation set, negative samples were generated once and held fixed, also comprising one sample per document. The training and validation sets are denoted as D_{train} and D_{val}, respectively.

The training routine was conducted over three epochs using the AdamW optimiser Loshchilov et al. [2017] and a cross-entropy loss function. We also used gradient clipping Zhang et al. [2019] and warmupm scheduling Kalra and Barkeshli [2024] for more stable and accelerated learning. Validation was performed every 500 training iterations to identify the optimal model based on performance metrics. The results of this validation are presented in Tab. 4.

Таблица 4: Validation of language model

Encoder and method	Accuracy	Precision-Macro	Recall-Macro	F1-micro
BERT + mean	0.887	0.761	0.699	0.720
BERT + attention	0.900	0.652	0.648	0.650
BERT + max pooling	0.914	0.717	0.647	0.667
BERT + endpoint	0.915	0.750	0.663	0.689
BERT + diff-sum	0.914	0.761	0.701	0.724
BERT + coherent	0.914	0.788	0.702	0.732
DeBERTa + mean	0.899	0.705	0.640	0.657
DeBERTa + attention	0.898	0.704	0.614	0.624
DeBERTa + max pooling	0.915	0.716	0.646	0.666
DeBERTa + endpoint	0.917	0.810	0.704	0.739
DeBERTa + diff-sum	**0.919**	**0.812**	**0.705**	**0.740**
DeBERTa + coherent	0.916	0.701	0.656	0.671
SpanBERT + endpoint	0.911	0.714	0.618	0.631

The best results are achieved by using DeBERTa and diff-sum (or endpoint) aggregation.

5.3 Validation of methods assessing difficulty

The validation of the considered methods was conducted in a manner analogous to the approach detailed in the article on gradient variance Agarwal et al. [2022]. Two distinct sets were defined: $D_{non-diff}$, comprising non-difficult spans, and D_{diff}, containing difficult spans. It is noteworthy that all four methods under examination, in common with numerous other approaches, are score-based. Consequently, the ROC-AUC metric was employed to evaluate the efficacy of each method in distinguishing between the pair $(D_{non-diff}, D_{diff})$. This metric can be interpreted as the probability that a non-difficult object will receive a lower difficulty score than a difficult one.

The validation set D_{val} was designated as the source of nominal non-difficult objects. Three variants of the set D_{diff} were constructed to represent nominal difficult objects: a noisy set D_{noise}, a subset D_{dd} from the ADE dataset featuring drug-dosage pairs, and a further subset D_{de} from ADE containing drug-effect pairs. The noisy set was generated by introducing Gaussian noise to the embeddings of spans from D_{val}; his noise was synthesised randomly from a normal distribution for each span prior to model training and validation. Spans from the ADE dataset were labelled with the "Other"tag, whereas the noisy spans retained their original corresponding labels.

The variance of gradients method Agarwal et al. [2022] was applied to all four sets (D_{val}, D_{noise}, D_{dd}, D_{de}) at 500-iteration intervals, concurrent with the validation of the language model. The variances computed at these checkpoints served as the difficulty scores.

The margin-based approach and the Gaussian method Lee et al. [2018] were applied to the best-performing model version following the completion of training. For the margin-based method, two alternative calculations for the difficulty score were investigated: $DiffScore(x) = -M(x)$, where difficult objects correspond to lower margins, and $DiffScore(x) = -|M(x)|$, where difficult objects are those exhibiting low absolute margin values. Here, x denotes a triplet $(T, (begin, end), tag)$ as previously defined. The Gaussian method utilised the Mahalanobis distance to compute the difficulty score. The mean μ_c and covariance matrix Σ were derived from the embeddings of the training spans prior to their processing by the classifier layer of the optimal model.

Results from the validation are provided in Table 5. The best results on ADE datasets as difficult sets are achieved with the Gaussian method. This is probably because the language models under consideration are able to separate well the vector representations of tokens and texts from different domains. Besides that, usage of BERT increased quality of difficulty assessing methods on ADE datasets for all aggregation methods.

If we consider Noisy dataset as difficult then Gaussian method has very higher ROC-AUC than other difficulty assessing methods for almost all models. However, when using the Gaussian method, the ROC-AUC is less than 0.5 if we use max pooling and coherent aggregations. Thus, usage of noise in difficulty assessing methods validation is less robust than using natural difficult objects.

Таблица 5: ROC-AUC for difficulty assessing methods applied to different difficult datasets

Encoder + method	Noisy dataset				Drug-dosage				Drug-effect			
	VoG	$-M(x)$	$-\|M(x)\|$	Gaus.	VoG	$-M(x)$	$-\|M(x)\|$	Gaus.	VoG	$-M(x)$	$-\|M(x)\|$	Gaus.
BERT + mean	0.563	0.584	0.590	0.999	0.737	0.766	0.802	0.948	0.694	0.698	0.732	0.933
BERT + attention	0.542	0.571	0.573	0.999	0.796	0.830	**0.846**	0.953	0.758	0.783	0.801	0.936
BERT + max pooling	0.560	0.594	0.593	0.456	0.855	0.826	0.820	0.958	0.863	0.841	**0.828**	0.950
BERT + endpoint	0.581	0.610	0.610	**1.000**	0.826	0.830	0.801	0.960	0.842	0.788	0.764	0.957
BERT + diff-sum	0.571	0.611	0.610	**1.000**	**0.869**	**0.865**	0.805	**0.967**	0.856	**0.863**	0.808	**0.965**
BERT + coherent	0.546	0.605	0.603	0.472	0.848	0.845	0.806	0.935	**0.867**	0.852	0.810	0.925
DeBERTa + mean	0.578	0.598	0.599	0.999	0.762	0.742	0.779	0.935	0.767	0.683	0.716	0.925
DeBERTa + attention	**0.617**	0.603	0.606	0.999	0.572	0.818	0.842	0.945	0.539	0.759	0.769	0.921
DeBERTa + max pooling	0.572	0.613	0.613	0.424	0.839	0.807	0.796	0.910	0.846	0.780	0.751	0.891
DeBERTa + endpoint	0.553	0.616	0.616	**1.000**	0.800	0.831	0.767	0.941	0.827	0.837	0.752	0.937
DeBERTa + diff-sum	0.580	**0.616**	**0.616**	**1.000**	0.845	0.809	0.781	0.913	0.835	0.831	0.789	0.920
DeBERTa + coherent	0.571	0.609	0.608	0.448	0.805	0.830	0.712	0.900	0.801	0.815	0.685	0.873
SpanBERT + endpoint	0.560	0.612	0.611	**1.000**	0.841	0.848	0.725	0.955	0.864	0.846	0.695	0.946

5.4 Correlations between difficulty assessing methods

In this experiment Spearman's rank correlation was calculated for each pair of difficulty assessing methods for each learned model. We considered only difficulty scores on objects from validation set D_{val}. These correlations are provided in Table 6.

Variance of gradients method and margin-based methods have higher correlations with each other than correlations with the Gaussian method which analyses directly vector representations of spans. Last mentioned correlations are quite large in most cases, which indicates the interpretability of the difficulty assessing methods relative to each other, however, these correlations still differ. The highest correlations between methods are observed when using average pooling and attention pooling as aggregation methods of language models.

Таблица 6: Correlations between different difficulty assessing methods for different models

Encoder + method	VoG			$-M(x)$			$-\|M(x)\|$			Gaussian		
	$-M(x)$	$-\|M(x)\|$	Gaus.	VoG	$-\|M(x)\|$	Gaus.	VoG	$-M(x)$	Gaus.	VoG	$-M(x)$	$-\|M(x)\|$
BERT + mean	**0.886**	**0.835**	**0.621**	**0.886**	0.936	0.560	**0.835**	0.936	0.521	**0.621**	0.560	0.521
BERT + attention	0.829	0.789	0.607	0.829	0.944	0.602	0.789	0.944	0.592	0.607	0.602	0.592
BERT + max pooling	0.725	0.689	0.526	0.725	0.958	0.398	0.689	0.958	0.391	0.526	0.398	0.391
BERT + endpoint	0.739	0.701	0.476	0.739	0.963	0.314	0.701	0.963	0.302	0.476	0.314	0.302
BERT + diff-sum	0.718	0.679	0.538	0.718	0.963	0.289	0.679	0.963	0.278	0.538	0.289	0.278
BERT + coherent	0.702	0.670	0.600	0.702	0.964	0.428	0.670	0.964	0.422	0.600	0.428	0.422
DeBERTa + mean	0.806	0.762	0.226	0.806	0.940	0.238	0.762	0.940	0.216	0.226	0.238	0.216
DeBERTa + attention	0.575	0.490	0.142	0.575	0.936	**0.604**	0.490	0.936	**0.602**	0.142	**0.604**	**0.602**
DeBERTa + max pooling	0.662	0.633	0.276	0.662	0.962	0.062	0.633	0.962	0.054	0.276	0.062	0.054
DeBERTa + endpoint	0.589	0.557	0.474	0.589	0.968	0.186	0.557	0.968	0.180	0.474	0.186	0.180
DeBERTa + diff-sum	0.727	0.697	0.409	0.727	0.968	0.081	0.697	0.968	0.071	0.409	0.081	0.071
DeBERTa + coherent	0.685	0.665	0.386	0.685	**0.970**	0.211	0.665	**0.970**	0.212	0.386	0.211	0.212
SpanBERT + endpoint	0.676	0.644	0.398	0.676	0.965	0.203	0.644	0.965	0.196	0.398	0.203	0.196

5.5 Interpretability of methods assessing difficulty

In this subsection, we present the spans from the validation set that exhibit the highest difficulty scores, as determined by various methods. Our focus is solely on natural spans derived from the original markups, excluding negative samples.

VoG In these examples, there are 3 typical situations when span objects has high VoG score. In the first case such span is poly-semantic ("Open") and we observe poly-semantic difficulty of object. In the second case object's difficulty is caused by over-fitting of the model: for example, model thinks, that "II"is a part of person's name instead of event. Third case is when labelling of span is controversial: "Lanka"may be both geo-political and geographical object. Thus, VoG scores are interpretable but many high scores are simply connected with model over-fitting.

Margin If we use $-M(x)$ as difficulty score then in most cases high scores are because model's over-fitting ("Greek "Time "Liechtenstein"), so margins are less interpretable. By the way, this method is still able to detect difficult spans: "People "vote".

Absolute value of margin In this method we observe that objects with lowest absolute values of margins are quite often correctly classified by the model but this method helps us detect named entities with difficult words and structure: "High Commissioner for Human Rights "EU Foreign Policy "Srebrenica"and others. Thus absolute values of margins are interpretable in their own way.

Gaussian method This method has connection with span embeddings anomalies. So there are two different cases: model's over-fitting ("Wimbledon "Canal") and difficulty of named entities' by themselves ("Please").

Therefore, all four methods are capable of finding difficult objects. It should be noted that they locate objects with varying degrees of difficulty; for example, the method of absolute value of margins enables the detection of spans with challenging names and structure. Additionally, the VoG method, the first approach of the margin method, and the Gaussian method are susceptible to overfitting when assessing the difficulty of objects.

5.6 Connection between difficulty and consistency

In this experiment we analysed relation between difficulty scores and confidence scores of spans in several ways.

Firstly, Spearman's rank correlation was calculated between difficulty scores and uncertainty scores on validation set for each difficulty assessing method and each model assuming that the inconsistency corresponds difficulty. Results are shown in Table 7. Since documents with multi-assessor markups are a minority, the correlations turned out to be quite low. By the way, margin-based approach methods have the highest correlations with uncertainty scores because of direct analysis of classifier's margins. Variance of gradient methods mostly correlates with inconsistency of spans when using DeBERTa and attention pooling, Gaussian method mostly correlates when using BERT with average or attention pooling.

It was mentioned, that there are few documents with several different markups so it was decided to make another experiment: we applied ROC-AUC to difficulty scores considering consistent objects as negative objects and inconsistent objects as positive objects. Thus we checked that inconsistent objects have on average higher difficulty scores than consistent objects. Results are shown in Table 8. Analogically to previous

experiment margin-based methods have the highest ROC-AUC. Besides that, we also observe that usage of BERT increases ROC-AUC of margin-based methods (it also increased correlations with uncertainty scores). But if we consider Variance of gradients and Gaussian method then the highest ROC-AUC of VoG corresponds to SpanBERT with endpoint aggregation while the highest ROC-AUC of DeBERTa correspond to DeBERTa with average pooling.

Finally, we also checked whether distributions of difficulty scores of consistent and inconsistent objects differ or not. We calculated p-values from the Kolmogorov-Smirnov test taking into account that small enough p-values allows to reject the hypothesis that difficulty scores of consistent and inconsistent objects are from the same distribution. This rejection occurred to be possible only in several situations when we consider application of margin-based methods to BERT model with max pooling, endpoint, diff-sum or coherent aggregations (p-values are about 1%). When we consider VoG or Gaussian method then there are few cases when p-values are only about 11-12%.

Таблица 7: Correlation between inconsistency scores and difficulty scores

Encoder and method	VoG	$-M(x)$	$-\lvert M(x)\rvert$	Gaussian
BERT + mean	0.000	0.013	0.009	0.014
BERT + attention	0.004	0.010	0.004	**0.015**
BERT + max pooling	0.003	**0.028**	**0.027**	0.008
BERT + endpoint	0.001	0.028	0.026	0.002
BERT + diff-sum	0.002	0.028	0.025	0.002
BERT + coherent	0.000	0.028	0.026	0.014
DeBERTa + mean	0.004	0.010	0.003	0.015
DeBERTa + attention	**0.016**	0.008	0.004	0.007
DeBERTa + max pooling	-0.003	0.012	0.009	0.007
DeBERTa + endpoint	-0.003	0.020	0.016	0.001
DeBERTa + diff-sum	0.004	0.021	0.017	0.003
DeBERTa + coherent	-0.001	0.020	0.016	0.000
SpanBERT + endpoint	0.012	0.016	0.015	0.006

Таблица 8: ROC-AUC applied to difficulty scores of consistent objects (as negative ones) and inconsistent objects (as positive ones)

Encoder and method	VoG	$-M(x)$	$-\lvert M(x)\rvert$	Gaussian
BERT + mean	0.503	0.628	0.581	0.634
BERT + attention	0.458	0.599	0.541	0.641
BERT + max pooling	0.530	**0.772**	**0.755**	0.573
BERT + endpoint	0.508	0.765	0.746	0.479
BERT + diff-sum	0.484	0.766	0.743	0.478
BERT + coherent	0.499	0.765	0.746	0.630
DeBERTa + mean	0.535	0.596	0.525	**0.647**
DeBERTa + attention	0.351	0.578	0.538	0.435
DeBERTa + max pooling	0.472	0.613	0.589	0.439
DeBERTa + endpoint	0.473	0.688	0.649	0.507
DeBERTa + diff-sum	0.542	0.706	0.660	0.475
DeBERTa + coherent	0.492	0.693	0.655	0.504
SpanBERT + endpoint	**0.613**	0.650	0.642	0.562

6 Conclusion

This article presents a pipeline for estimating the model-specific difficulty of a tagged text span, with an ultimate goal of enhancing the performance of machine learning models. Within this context, the difficulty of an object is defined as the complexity a machine learning model encounters when processing it for a specific task. An established language model architecture, recognised for its high-quality vectorisation capabilities, was adapted for this purpose. A language model was subsequently trained and evaluated using our approach. The efficacy of the task-specific difficulty assessment methods was then analysed through multiple approaches;

Таблица 9: The p-values from the Kolmogorov-Smirnov test for checking the hypothesis that difficulty scores of consistent and inconsistent objects are from the same distribution

Encoder and method	VoG	$-M(x)$	$-\|M(x)\|$	Gaussian
BERT + mean	0.940	0.399	0.355	0.386
BERT + attention	0.960	0.307	0.264	0.224
BERT + max pooling	0.838	**0.012**	**0.009**	0.392
BERT + endpoint	0.837	0.028	0.024	0.677
BERT + diff-sum	0.865	0.049	0.044	0.835
BERT + coherent	0.939	0.025	0.021	**0.128**
DeBERTa + mean	0.900	0.822	0.783	0.389
DeBERTa + attention	0.120	0.606	0.506	0.456
DeBERTa + max pooling	0.702	0.736	0.703	0.869
DeBERTa + endpoint	0.751	0.272	0.447	0.861
DeBERTa + diff-sum	0.718	0.132	0.243	0.783
DeBERTa + coherent	0.781	0.415	0.404	0.966
SpanBERT + endpoint	**0.113**	0.125	0.089	0.576

these included calculating the ROC-AUC metric across various pseudo-difficult datasets and conducting a qualitative examination of objects assigned the highest difficulty scores.

The Gaussian approach Lee et al. [2018] was found to achieve the highest ROC-AUC score. In contrast, the margin method, which utilises absolute values to derive difficulty scores, demonstrated greater interpretability. Furthermore, our analysis identified variations in the perceived difficulty of individual objects, which were contingent upon the specific assessment method employed.

Besides that, DeBERTa with endpoint and diff-sum span aggregations achived the highest ROC-AUC scores across all methods which indicates a connection between the quality of models and the quality of the difficulty assessment of spans relative to its work.

Additionally, a comparison was made between different types of datasets containing difficult examples. The use of datasets comprising texts on diverse topics proved to be a more stable and reliable basis for the application of various difficulty assessment techniques. Conversely, the qualitative recognition of difficulty within a noisy dataset was found to necessitate direct analysis of the span embeddings themselves.

Then we checked interpretability of difficulty assessing methods. We checked that methods correlate with each other but defined that degrees of correlations differ - Gaussian method have lowest correlation scores because it analyses span embeddings directly unlike other methods.

We also considered spans with the highest difficulty scores and discovered that absolute margin method is the most interpretable while other methods sometimes return high difficulty scores due to overfitting of models. Method of absolute margins finds spans with difficult words and phrases, variance of gradients method often finds spans with poly-semantic meaning.

Finally we discovered whether difficulty scores and consistency of spans have connection or not calculating rank correlations, ROC-AUC and p-values from the Kolmogorov-Smirnov test. Margin-based methods assess difficulty scores that have the highest connection with consistency of spans and it is best demonstrated when using max pooling aggregation with BERT encoder. Variance of gradients method and Gaussian method have high connection between difficulty and consistency infrequently when using only certain models and aggregation techniques.

Список литературы

Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. Dissecting sample hardness: A fine-grained analysis of hardness characterization methods for data-centric ai. In The Twelfth International Conference on Learning Representations, 2024.

David Tax and Robert Duin. Support vector data description. Machine Learning, 54:45–66, 01 2004. doi:10.1023/B:MACH.0000008084.60811.49.

Ertunc Erdil, Krishna Chaitanya, Neerav Karani, and Ender Konukoglu. Task-agnostic out-of-distribution detection using kernel density estimation. In Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis: 3rd International Workshop,

UNSURE 2021, and 6th International Workshop, PIPPI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 3, pages 91–101. Springer, 2021.

Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10, pages 36–51. Springer, 2017.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. ArXiv, abs/1807.03888, 2018. URL https://api.semanticscholar.org/CorpusID:49667948.

Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906.02694, 2019.

Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S Yoon. Rapp: Novelty detection with reconstruction along projection pathway. In International Conference on Learning Representations, 2019.

Thomas Schlegl, Philipp Seeböck, Sebastian Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. pages 146–157, 03 2017. ISBN 978-3-319-59049-3. doi:10.1007/978-3-319-59050-9_12.

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10368–10378, 2022.

Yijin Zhou and Yuguang Wang. Grod: Enhancing generalization of transformer with out-of-distribution detection. arXiv preprint arXiv:2406.12915, 2024.

Emmeke Aarts, Matthijs Verhage, Jesse V Veenvliet, Conor V Dolan, and Sophie Van Der Sluis. A solution to dependency: using multilevel analysis to accommodate nested data. Nature neuroscience, 17(4):491–496, 2014.

Archil Maysuradze, Olga Rink, Artem Fedorov, Andrey Tabachenkov, and Konstantin Vorontsov. Does annotating multi-spans improve classification in considerable text collections? In 2024 10th International Conference on Systems and Informatics (ICSAI), pages 1–6, 2024. doi:10.1109/ICSAI65059.2024.10893762.

Olga Rink, Viktor Lobachev, and Konstantin Vorontsov. Detecting human values and sentiments in large text collections with a context-dependent information markup: A methodology and math. In International Conference on Human-Computer Interaction, pages 372–383. Springer, 2024.

K. Vorontsov, I. Gladchenko, V. Lobachev, A. Mamontova, O. Rink, and N. Shabelskaya. Developing an open interdisciplinary classifier of human values by means of annotating multi-fragments. Vestnik of Saint Petersburg University. International Relations, 18:38–62, 2025. doi:10.21638/spbu06.2025.103.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. In Named Entities: Recognition, classification and use, pages 3–28. John Benjamins publishing company, 2009. URL https://www.kaggle.com/datasets/naseralqaydeh/named-entity-recognition-ner-corpus.

Anton Golubev, Nicolay Rusnachenko, and Natalia Loukachevitch. RuSentNE-2023: Evaluating entity-oriented sentiment analysis on russian news texts. In Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue" (arxiv:2305.17679, 2023. URL https://github.com/dialogue-evaluation/RuSentNE-evaluation.

Alena Fenogenova, Artem Chervyakov, Nikita Martynov, Anastasia Kozlova, Maria Tikhonova, Albina Akhmetgareeva, Anton Emelyanov, Denis Shevelev, Pavel Lebedev, Leonid Sinev, Ulyana Isaeva, Katerina Kolomeytseva, Daniil Moskovskiy, Elizaveta Goncharova, Nikita Savushkin, Polina Mikhailova, Anastasia Minaeva, Denis Dimitrov, Alexander Panchenko, and Sergey Markov. MERA: A comprehensive LLM evaluation in Russian. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9920–9948, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.534. URL https://aclanthology.org/2024.acl-long.534.

Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. Extraction of potential adverse drug events from medical case reports. Journal of biomedical semantics, 3:1–10, 2012. URL https://huggingface.co/datasets/ade-benchmark-corpus/ade_corpus_v2.

Natalia Loukachevitch, Ekaterina Artemova, Tatiana Batura, Pavel Braslavski, Vladimir Ivanov, Suresh Manandhar, Alexander Pugachev, Igor Rozhkov, Artem Shelmanov, Elena Tutubalina, et al. Nerel: a russian information extraction dataset with rich annotation for nested entities, relations, and wikidata entity links. Language Resources and Evaluation, pages 1–37, 2023. URL https://github.com/nerel-ds/NEREL.

Denis Gordeev, Adis Davletov, Alexey Rey, Galiya Akzhigitova, and Georgiy Geymbukh. Relation extraction dataset for the russian language. In Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog"[Komp'iuternaia Lingvistika i Intellektual'nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii "Dialog"]. Russian State University For The Humanities Moscow, Russia, 2020. URL https://github.com/denis-gordeev/rured2.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreferencefor scientific knowledge graph construction. In Proc. Conf. Empirical Methods Natural Language Process. (EMNLP), 2018. URL https://nlp.cs.washington.edu/sciIE/.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. arXiv preprint arXiv:1911.10422, 2019. URL https://huggingface.co/datasets/sem_eval_2010_task_8.

Khiem H Le, Tuan V Tran, Hieu H Pham, Hieu T Nguyen, Tung T Le, and Ha Q Nguyen. Learning from multiple expert annotators for enhancing anomaly detection in medical image analysis. IEEE Access, 11: 14105–14114, 2023.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. Transactions of the association for computational linguistics, 8:64–77, 2020.

Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In ECAI 2020, pages 2006–2013. IOS Press, 2020.

Shubham Toshniwal, Haoyue Shi, Bowen Shi, Lingyu Gao, Karen Livescu, and Kevin Gimpel. A cross-task analysis of text span representations. In Proceedings of the 5th Workshop on Representation Learning for NLP, pages 166–176, 2020.

Tom Fawcett. An introduction to roc analysis. Pattern recognition letters, 27(8):861–874, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pages 4171–4186, 2019.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In International Conference on Learning Representations, 2021.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. 2016.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101, 5:5, 2017.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In International Conference on Learning Representations, 2019.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. Advances in Neural Information Processing Systems, 37:111760–111801, 2024.