Multi-Attribute Relation Extraction (MARE): Simplifying the Application of Relation Extraction

Lars Klöser[1], Philipp Kohl[1], Bodo Kraft[1] and Albert Zündorf[2]
[1]*FH Aachen, University of Applied Sciences, Germany*
[2]*University of Kassel, Germany*

Keywords: Natural Language Processing, Natural Language Understanding, Information Extraction, Relation Extraction, Joint Relation Extraction, Event Extraction.

Abstract: Natural language understanding's relation extraction makes innovative and encouraging novel business concepts possible and facilitates new digitilized decision-making processes. Current approaches allow the extraction of relations with a fixed number of entities as attributes. Extracting relations with an arbitrary amount of attributes requires complex systems and costly relation-trigger annotations to assist these systems. We introduce **m**ulti-**a**ttribute **r**elation **e**xtraction (MARE) as an assumption-less problem formulation with two approaches, facilitating an explicit mapping from business use cases to the data annotations. Avoiding elaborated annotation constraints simplifies the application of relation extraction approaches. The evaluation compares our models to current state-of-the-art event extraction and binary relation extraction methods. Our approaches show improvement compared to these on the extraction of general multi-attribute relations.

1 INTRODUCTION

Small and medium-sized enterprises (SMEs) increasingly recognize the potential of *natural language understanding* and *relation extraction* to digitalize processes and develop novel software products. Many product visions include the extraction of variable-sized sets of concept mentions as relations from texts where existing data models define a set of potential attributes per relation. But most current approaches focus on the extraction of binary relations. E.g., the number of many thousand biomedical scientific publications per week yielded the successful automation of knowledge discovery (Tsujii et al., 2011; Kim et al., 2011a; Kim et al., 2011b). In contrast to many other domains, binarity's structural constraint seems to be reasonable due to cause-effect-relationships. The extraction of more complex semantic relations currently requires the construction of sophisticated systems based on binary classifications. The field of event extraction covers such approaches. Events are multi-attribute relations with so-called trigger annotations. For example, in the following message about a traffic obstruction: *A1 between Köln-Mühlheim and Köln-Dellbrück objects on the road, both directions closed*. According to (Consortium, 2005) *closed* triggers the event but provides no event-specific information. Attributes assigned to such triggers build event relations. The central role of trigger annotation results in high-quality requirements and an increased annotation effort.

This research introduces multi-attribute relation extraction (MARE), a novel problem definition that aims to simplify the application relation extraction approaches in practice. *Multi-attribute relations*:

- have a well-known set of potential roles for attributes,
- make no assumptions on attributes' multiplicity building a relation instance, and
- do not rely on the trigger concept, indicating one's relation presence.

We introduce a sequence tagging and a span labeling approach to recognize entities and extract multi-attribute relations between them in a joint model. We analyze our approaches' performance on the *SmartData* corpus (Schiersch et al., 2018). This corpus is the only available resource for relation extraction on German texts. This corpus's annotations include named entities and multi-attribute relations between those. We publish all data and source code connected to the research in a GitHub repository[1]. Our main

[1]https://github.com/MSLars/mare

Klöser, L., Kohl, P., Kraft, B. and Zündorf, A.
Multi-Attribute Relation Extraction (MARE): Simplifying the Application of Relation Extraction.
DOI: 10.5220/0010559201480156
In *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications (DeLTA 2021)*, pages 148-156
ISBN: 978-989-758-526-5

contributions can be summarized as follows:

- We formalize multi-attribute relation extraction and introduce two problem-specific approaches.
- We show that the non-trigger-based approaches have in general a better performance on the multi-attribute relations in the SmartData corpus.
- We provide the first reproducible evaluation of a non-binary relation extraction approach on a German corpus.

2 RELATED WORK

Relation extraction investigates the mutual relation between named entities in texts to transfer the unstructured information into predefined schemas. Most benchmark datasets consider only binary relations (Mintz et al., 2009; Hendrickx et al., 2010).

Traditional *binary relation extraction* approaches use *part of speech tagging*, *dependency parsing*, and further steps to calculate input representations for machine learning models (Xu et al., 2013). Current state-of-the-art models use transformer networks to calculate highly contextualized representations of binary relation candidates and combine these with specialized decision layers in a combined neural network (Li and Tian, 2020; Eberts and Ulges, 2019).

As an extension to binary relation extraction, the field of n-ary relation extraction aims to detect relations with a fixed number of n arguments. (Peng et al., 2017) extended *recurrent neural networks* to efficiently include syntactic dependency edges to build contextualized relation representations. (Lai and Lu, 2021) presented a transformer-based approach for the same experimental setup. Both focus on 3-ary relations.

Binary and small n-ary relation extraction approaches often enumerate sets of predicted entities as a root for building relation candidates. We extract relations with an arbitrary number of attributes avoiding such an enumeration to prevent a combinatorial explosion.

To extend the fixed-size constraint, the field of *event extraction* defines events as multi-attribute relations with one necessary trigger attribute. The trigger indicates the presence of an event. Other entities can be assigned to single triggers to form event relations (Consortium, 2005; Aguilar et al., 2014). Event extraction approaches rely on this trigger annotations (Xiang and Wang, 2019). All entities assigned to one trigger form a multi-attribute relation. This reduces the problem to a sequence of binary relation classifications.

Traditionally, real-world relation extraction systems extract entities and their relationships in a processing pipeline. Such systems suffer from error propagation. The field of *joint relation extraction* investigates models that extract entities and relations in a single model. A common way to build a joint model is to share the embedding layer across multiple downstream tasks. (Wadden et al., 2019) introduced a system that shares embeddings to extract named entities, builds binary relation candidates, and classifies the relation between those. (Zheng et al., 2017; Liu et al., 2019) introduce sequence label schemes to explicitly extract attributes and their relations in a single classification step. Our models similarly extract more complex structures without an enumeration of relation candidates. This avoids a combinatorial explosion for MARE. We apply novel transformer network to receive contextualized text-embeddings (Devlin et al., 2019; Clark et al., 2020).

(Schiersch et al., 2018) introduced the SmartData Corpus for relation extraction on German texts. The annotated relations contain a various number of mandatory and optional arguments. Section 3 analyzes the corpus in great detail. The original paper includes results of the relation extraction system DARE (Xu et al., 2013). This evaluation considers only mandatory attribute roles for each relation. We also consider optional attributes and analyze results on a more sophisticated problem setting. (Roller et al., 2018) investigates the extraction of named entities and binary relations from German clinical reports. Their corpus is unpublished.

3 DATA ANALYSIS

We train and evaluate our approaches on the *SmartData*[2] corpus (Schiersch et al., 2018), a German corpus provided by the DFKI[3]. The corpus contains manually annotated traffic and industry entities and relations in News, RSS feeds, and tweets.

The third corpus version contains 19,116 entities and 1,264 relations in 2,322 documents with 141,344 words in total[4]. Table 1 shows the train-test-split provided by SmartData.

The *inter-annotator agreement* is on a moderate level (Viera et al., 2005) with Cohen's kappa coefficient for entities of 0.58 and 0.51 for relations.

[2]https://github.com/DFKI-NLP/smartdata-corpus

[3]Deutsches Forschungszentrum für Künstliche Intelligenz (Translation: German Research Center for Artificial Intelligence)

[4]Numbers differ from the original paper due to different versions

Table 1: The SmartData corpus' train-test-split with the number of relations and the fraction of documents in each subset.

Data set	Documents	Relations	Ratio
Training	1,864	1,007	0.8
Validation	228	129	0.1
Test	230	128	0.1
Sum	2,322	1,264	1

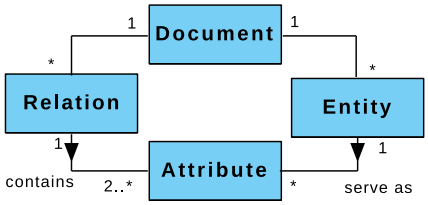

Figure 1: Structure of instances in the SmartData corpus. Documents contain relations and entities. A relation has at least two mandatory attributes. Each attribute has an entity mention. Entities may function as attributes in zero or multiple relations. E.g., the entity *Location* may serve as an attribute in *Accident* and *Obstruction* at the same time.

DFKI describes their preprocessing steps in (Schiersch et al., 2018) and GitHub.

Figure 1 illustrates the data meta-model. Note that a relation can have a variable number of attributes and is not limited to a fixed number. For each attribute's role a fixed set of entity types may fit: E.g., entity types such as *Location-Street*, *Location-City* or *Location-Route* can serve interchangeably as an attribute with role *Location*.

In the following, we explain the key characteristics of the SmartData corpus.

Relations. The corpus provides 15 relation types with two mandatory and arbitrary optional attributes. Figure 3 illustrates the relation's and the attribute's distribution.

Entities. SmartData provides 16 fine-grained entity types. For a full list, see (Schiersch et al., 2018). We introduce *explicitness* as metric to demonstrate that only a few entity types are a strong indicator for a relation (cf. *Jam Length* or *Position* in Figure 3). Therefrom, MARE models have to learn a combined view of entity compounds.

Variable Number of Relation Attributes. Each relation contains at least one of each mandatory attributes. They may or may not contain further optional attributes. The example for *Arguable differences* in Figure 5 shows an RSS feed with an *Obstruction* relation. Only *trigger* and *location* attributes are mandatory. *StartLoc* and *EndLoc* are optional attributes.

Unbalanced. Unbalanced datasets raise the challenge to learn the essential structure for underrepresented data points and the other richer data points (Mountassir et al., 2012). The dataset is unbalanced in terms of relations as of attributes as well (cf. Figure 3): *Traffic Jam* occurs approximately 10 times more often than *Spin Off*. While Spin Off's attributes frequencies are quite equal, Traffic Jam's attributes show a difference between the attribute's distribution, which corresponds to mandatory and optional attributes.

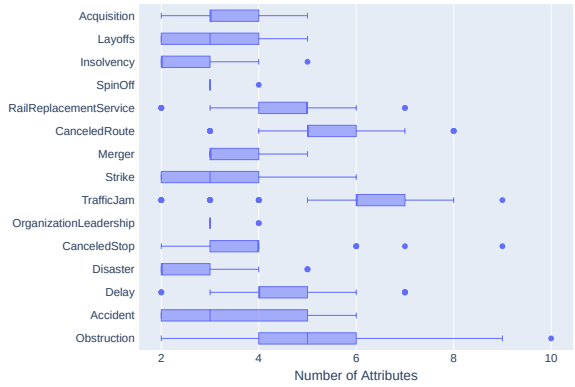

Figure 2: Boxplot illustrating the distribution of the number of attributes per relation. E.g., the number of *Obstruction's* attributes ranges from two to ten while other relations like *Insolvency* do not show the same variance. Dots indicate outliers.

Improper Triggers. Other event extraction corpora's triggers are strictly defined to one single mandatory token or span due to its essential role as relation indicator (Consortium, 2005; Aguilar et al., 2014). SmartData does not follow these constraints: the triggers are optional and not bound to consecutive tokens or any specific lemma or part of speech. Thus, this corpus impedes the application of current event extraction approaches due to their assumption of one existing trigger token/span.

Relations Share Entities. Multiple relations can occur in one document. The corresponding relation's entities dot not have to be disjointed: e.g., *Traffic Jam* and *Obstruction* likely appear together and sharing *location* attributes.

Different Register of Language. SmartData uses different data sources leading to different distributions, and patterns models have to learn. While news articles are continuous and grammatically valuable text, Twitter and RSS feeds are often sentence fragments.

The SmartData corpus provides relations with a variable amount of attributes and without a regular trigger definition, making the corpus fit into the MARE definition. Modifications are necessary to apply current relation or event extraction approaches.

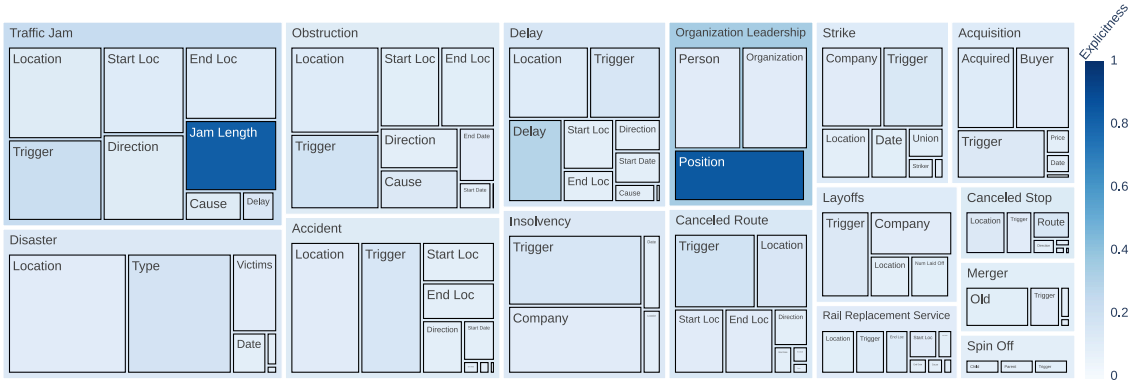

Figure 3: Distribution of relations and attributes. The rectangles' sizes are proportional to the relation or attribute frequency. The explicitness is the quotient of the attributes' frequency and the total number of entities with an entity type suitable for the specific attribute role. The metric indicates how reliably an entity type indicates a relation attribute.

4 MARE

This section formally introduces the concept of multi-attribute relation extraction and introduces two MARE approaches. We describe our evaluation methodology, which includes the adaptation of an event and binary relation extraction approach. We compare both against the MARE approaches.

4.1 Definition

For a given text $t = (t_1, \ldots, t_n)$ with n tokens,

$$S = \{(t_i, \ldots, t_j) \mid \text{for all } i, j \in \{1, \ldots, n\}, i \leq j\}$$

denotes the set of all text-spans. Let L be a set of relation labels and A_l be a set of attribute roles for each relation label $l \in L$. The task is to predict a relation set R for a given text t. Each relation instance $r \in R$

$$r = (l, \{\alpha_i \mid \text{for all } i \in \{1, \ldots, m\}\})$$

consists of a relation label l and a variable number of $0 < m \leq |S|$ attributes

$$\alpha_i = (s, a) \in S \times A_l \text{ for all } i \in \{1, \ldots, m\}.$$

Each span $s \in S$ can contribute to at most one attribute in each relation $r \in R$. But a span can contribute to attributes in multiple relations. We explicitly allow relations with one attribute. We denote text-spans s_{ij} with i, j as start and end indices. Further,

$$A = \bigcup_{l \in L} A_l$$

is the set of all attribute roles.

The formal definition makes no difference between mandatory and optional attributes as in Section 3. We still use this distinction for the model evaluation since a higher frequency of an attribute role implies a better extraction performance.

4.2 Approaches

All approaches except the baselines use transformer networks as contextualized embedders. Such networks compute contextualized representations with a combination of multiple *self-attention* and *feed-forward-layers*. They are trained in an unsupervised fashion (Devlin et al., 2019). We apply a german version of ELECTRA[5] (May and Reißel, 2020). Its pretraining tasks focus on the models' ability to describe the semantic structure of texts (Clark et al., 2020). All approaches use the version of the Adam optimization algorithm with weight decay introduced in (Loshchilov and Hutter, 2019).

In the following, we use the definitions introduced in Section 4.1.

4.2.1 Sequence Tagging

The unknown number of attributes in MARE requires models that do not need to enumerate all relation candidates. (Zheng et al., 2017) introduced a tagging scheme to formulate binary relation extraction as a sequence tagging problem.

$$T = \{b, i\} \times L \times A \cup \{o\}$$

describes our tag-set. Tags that start with b and i mark tokens as the beginning or inner part of an entity. For the resulting entity spans, the label $l \in L$ determines the relation, and $a \in A_l$ determines the attribute role for a relation determined by l. o marks tokens that do not belong to an attribute.

From each tagged token sequence, we extract a set of incoherent relation attributes. These attributes

[5]https://huggingface.co/german-nlp-group/electra-base-german-uncased

are summarized to relation instances by their relation label.

The embedded and contextualized input sequence is the input for a feed-forward layer, which maps them to label probabilities. A *conditional random field* determines the loss and the most probable label sequence. (Huang et al., 2015) describes the details of conditional random fields for sequence tagging.

Our sequence tagging model avoids the enumeration of all potential relation candidates. However, this also leads to the two following limitations:

1. **Shared Attributes Across Relations.** Multiple relations may have attributes with shared text spans. Our tagging scheme can only assign each span to at most one relation.
2. **Multiple Relations with the Same Label.** A sample could have multiple relations with the same relation label. For example, two accident descriptions in one sample. A grouping based on the label leads to a single relation instead of multiple instances.

We introduce a layer of business logic to deal with such situations. In the case of shared attribute spans across various relations, we check whether the current relations have any missing mandatory attributes. If so, we search for attribute types indicating shared arguments. If such an attribute is within a *maximum relation width*[6], we use it to complete the relation.

In the following, we assume relation attributes to be sorted by their span indices. To handle multiple relations with the same label in one sample, we split a grouped relation $\alpha_1, ..., \alpha_n$ at an index $i < n$ if the subsets $\alpha_1, \ldots, \alpha_i$ and $\alpha_{i+1}, \ldots, \alpha_n$ contain all mandatory attributes and the distance between α_i and α_{i+1} exceeds the maximum relation width.

4.2.2 Span Labeling

Our second approach is motivated by (Liu et al., 2019). They applied a sequence labeling approach instead of sequence tagging. Labeling allows the assignment of multiple attribute labels to each text-span. We modify this approach and predict a relation-attribute label for each possible text-span in a given sample. As our sequence tagging approach, this approach does not need to enumerate all relation candidates and resolves the limitation of shared attributes across relations.

Let $T = L \times A$ be a set of labels indicating the relation label and attribute role for a given text-span.

[6]The maximum relation width is a hyperparameter and determined by a hyperparameter search. The GitHub repository contains the search configuration and final values.

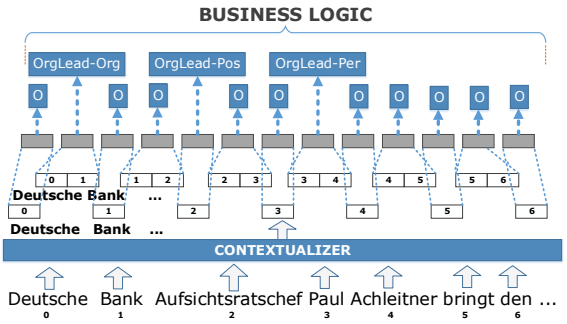

Figure 4: Illustration of the span labeling approach. The input sequence is embedded and contextualized. Each text-span within a *maximum span width* (2 in this example) is transformed to a fixed-length representation labeled with a relation-label and argument-role combination. Finally, the business logic groups attribute to relations.

The model predicts a probability $P(t|s)$ for each label $t \in T$ and each span $s \in S$. A *maximum span width* hyperparameter defines the maximum number of tokens per span in S. We apply a *binary cross-entropy loss* function, which allows the assignment of multiple labels per span.

Figure 4 illustrates our model architecture. The span representations are computed with a self-attention-based module from AllenNLP[7]. For a given text with length n we compute contextualized embeddings $(c_1, \ldots, c_n)$ of dimension d. For each span s_{ij}, we have $j - i + 1$ embeddings $(c_i, \ldots, c_j)$. To get a fixed-length span representation of dimension d, we calculate a linear combination of these embeddings. A parameter matrix $M \in \mathbb{R}^{d \times 1}$ calculates global attention scores $a_i = c_i \cdot M$ for all $i \in \{1, \ldots, n\}$. These are used to calculate weigths $w_i, \ldots, w_j$ for a span s_{ij}, with

$$w_k = \frac{e^{a_k}}{\sum_{i \le l \le j} e^{a_l}} \text{ for all } k \in \{i, \ldots, j\}.$$

The softmax function ensures that the weights for each span sum-up to 1. The final span representations are a linear combination of these weights and the embeddings.

A *feed forward layer* in combination with the element-wise *sigmoid* function computes the label probabilities for each span. Similar to the previous approach, this leads to a set of grouped relation instances. We apply the same business logic as in Section 4.2.1 since the limitation of *multiple relations with the same label* remains.

[7]http://docs.allennlp.org/main/api/modules/span_extractors/self_attentive_span_extractor/

4.2.3 Event Extraction

To apply event extraction approaches, we need to specify an event trigger for each multi-attribute relation. As Section 3 shows, some instances in the SmartData corpus have no such annotations. If a relation definition has no mandatory trigger attribute, we defined one mandatory attribute type for each relation as the trigger. In the case of multiple and non-conjunct trigger spans, we select the first span as the trigger. We did not apply more complex logic since the set of relations with multiple triggers (78 of 1264) is relatively small. The first error situation in Section 4.2.1 is unsolved if relations share triggers. Other attributes can be shared across relations.

We apply Dygie++[8] as event extraction approach. As (Wadden et al., 2019) describes, Dygie++ uses contextualized span representations, similar to Section 4.2.2. Trigger detection and attribute disambiguation use this shared span representations.

4.2.4 Binary Relation Extraction

Many binary relation extraction approaches classify all possible pairs of entities as relation candidates, as SpERT (Eberts and Ulges, 2019). In combination with multi-class labeling, this solves both error situations in Section 4.2.1.

We apply SpERT to extract binary relations from 1,717 of 1,864 samples in the train split that contain relations with exactly two mandatory attributes. The next section introduces various evaluation strategies. We introduce a *binary relation extraction* strategy to compare the performance of SpERT against all other approaches on the subset of valid binary relations.

4.3 Experimental Setup

We used AllenNLP (Gardner et al., 2017) and Pytorch to implement the sequence tagging and span labeling approach. Our GitHub repository contains modified versions of Dygie++ and SpERT. These modifications were necessary to integrate both approaches into our experimental infrastructure.

Our GitHub repository contains a summary of all hyperparameters and their values in the final models. All hyperparameters were determined with Optuna[9]. We applied 50 optimization trials for each model. We fixed the learning rate for the transformer network's embedding layer to $5 \cdot 10^{-5}$ and 10^{-3} for all other network components. We selected a batch size of 6 for all MARE approaches, for SpERT and Dygie++ a batch size of 1.

We use various evaluation strategies to analyze the predictions. The strategies aim to reflect the challenges of MARE on different levels of complexity.

- **Attribute Recognition (AR).** The evaluation is made attribute wise. An attribute is considered correct if its boundaries, relation label, and attribute role are predicted correctly while not considering the grouping to a relation.
- **Classification (Cl).** A prediction is correct if the predicted label matches a gold relation label.
- **Mandatory Relation Extraction (MRE).** A prediction is correct if all mandatory attributes and the relation label match against the gold annotation. Thus, the grouping of the mandatory attributes to one relation is essential.
- **Complete Relation Extraction (CRE).** Measures the model's capability of extracting the relation with all attributes as a whole. Thus, a prediction is considered correct if the model extracts all attributes and groups them correctly into a relation with the right relation label.
- **Binary Relation Extraction (BRE).** This is the MRE strategy on the subset of samples that contains only relations with exactly two mandatory arguments. This strategy allows a comparison between SpERT and all other approaches.

We include the baseline (DARE) from (Schiersch et al., 2018), which focuses on the mandatory arguments and uses gold entity annotations. Our own baseline is a modification of the sequence tagging approach. We replace the pretrained transformer network with a combination of GloVe[10] word vectors (Pennington et al., 2014) and character level CNN as embedding layer. A Bi-GRU layer contextualizes the inputs.

Our computational setup contains two nodes with Intel Xeon Platinum 8168 CPUs, Nvidia Quadro P5000 GPUs with 16 GB RAM, and Ubuntu 18.04 OS. A hyperparameter search took approximately 24 hours.

5 RESULTS

The metrics in Table 2 measure different capabilities necessary to extract all attributes, their roles, and the relation label in combination. In general, as the requirements of the metrics increase, the metrics' values decrease.

[8]https://github.com/dwadden/dygiepp

[9]https://optuna.org/

[10]https://deepset.ai/german-word-embeddings

The span labeling approach increases the event extraction AR results by 0.06 and the Cl results by 0.04 in the F1 scores. We observe that both MARE approaches perform better than Dygie++ on the complete dataset. The similar MRE score and the increased AR and CRE scores indicate that MARE models extract optional and potentially less-frequent arguments more reliably. A reduction to the subset of documents with exactly two mandatory arguments leads to higher general scores but a much higher increase for Dygie++ than for the MARE models. Both observations indicate that model architectures with fewer structural assumptions are better suitable for the corpus' unique characteristics as described in Section 3.

Table 2: Model evaluation on the test set based on different strategies, see Section 4.3. Precision, Recall and F1 score serve as comparison metrics.

Model		AR	Cl	MRE	CRE	BRE
MARE	F1	.66	.76	.45	**.30**	.47
Seq. Tag.	P	.66	.73	.43	.28	.44
	R	**.66**	.80	**.48**	**.31**	.49
MARE	F1	**.70**	**.80**	**.47**	.29	.49
Span Lab.	P	**.75**	**.80**	.47	**.29**	.49
	R	.65	**.80**	.47	.29	.49
	F1	.64	.76	.46	.25	**.53**
Dygie++	P	.63	.77	.47	.26	.55
	R	.65	.74	.45	.25	**.52**
	F1	-	-	-	-	.51
SpERT	P	-	-	-	-	**.57**
	R	-	-	-	-	.45
MARE	F1	.60	.68	.39	.26	.41
Baseline	P	.66	.68	.40	.26	.40
	R	.55	.67	.39	.26	.41
	F1	-	-	.28	-	-
DARE	P	-	-	**.53**	-	-
	R	-	-	.19	-	-

The difference between the MARE baseline and both MARE approaches shows the positive effect of pretrained transformer networks. Despite the general weaker performance of DARE, which uses an automatically selected rule-set, the original benchmark has the highest MRE precision score. This indicates a high certainty in extracted relations and a high number of false negatives because of the low recall score.

The evaluation of SpERT shows a clear improvement compared to our MARE baseline. SpERT performs also better than our MARE models on BRE.

Table 3 shows how trigger annotations effect MARE models' performance. Evaluations on the reduced set of non-trigger attributes of MARE models trained both with or without trigger annotations

Table 3: Comparison of approaches trained with and without trigger annotations. We exclude trigger entities from the score computations.

Model	AR F1	MRE F1
Seq. Tag. with Trigger	.64	.49
Seq. Tag. without Trigger	.64	.51
Span Lab. with Trigger	.67	.52
Span Lab. without Trigger	.66	.54
Dygie++. with Trigger	.62	.53

do not show a significant difference in AR and MRE scores. Our models' performance without trigger annotations is comparable to state-of-the-art event extraction on multi-attribute relations from the SmartData corpus. The MARE models AR score is better than Dygie++'s. That proves MARE's ability to extract optional and less-frequent attributes.

Compared to Table 2 the AR scores decrease, indicating that the models extract trigger attributes reliably. Without trigger attributes many single-attribute relations remain. This simplifies the MRE task and causes the increased MRE scores.

Since relation extraction is a task on a high semantic level and SmartData's gold annotations contain a certain degree of inconsistency, we provide a manual error analysis to understand our models' prediction characteristics better.

5.1 Error Inspection

We conducted a manual comparison of the differences between the gold annotations and the models' predictions. The following error-equivalence-classes emerge from our manual inspection. All examples mentioned in the following enumeration refer to Figure 5.

1. **Arguable Differences.** The models' predictions are often reasonable even if the gold data contain divergent annotations. The example shows an *Obstruction* relation, in which the *marathon* represents the *Obstruction-Cause*. The gold annotation does not reflect this circumstance. Arguable differences indicate that our models learned certain semantic concepts. Some generalizations in the predictions lead to false positives lowering the evaluation metrics.
2. **Semantic Depending Relation Classes.** Some relation classes, such as *Accident* and *Obstruction* have a strong semantic relationship. Therefore, instances of these relations are often nested and share entity spans as attributes. The annotations of these shared attributes are often flawed. The example shows an *Obstruction* caused by a *Disas-*

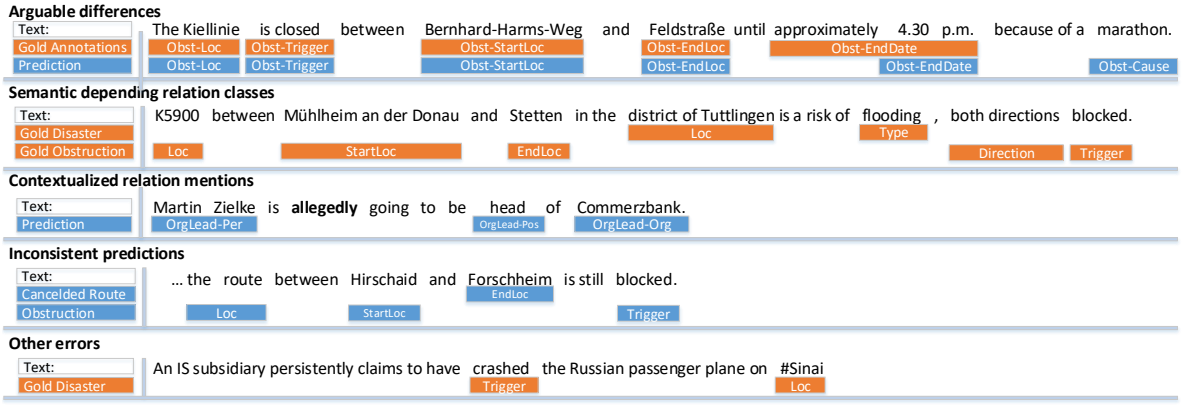

Figure 5: Examples for error classes. Colored boxes indicate the relations and their attributes. The attributes role is textually annotated. Predictions made by the span labeling approach.

ter. The gold annotation contains two separate relations and does not express this dependency. The trigger of the *Disaster* could also be interpreted as the cause of the *Obstruction*. This distinction is challenging for the models.

3. **Contextualized Relation Mentions.** Many supposed relation instances appear in a presuming context. Words like *allegedly* indicate assumptions rather than facts. The example shows a presumption about an *Organization Leadership*. In many of these cases, the models predicted relation instances.
4. **Inconsistent Predictions.** The example shows an *Obstruction* relation, where the model predicted all roles correctly. The relation label for the *End Location* belongs to a similar semantic relation. If the missing attributes are not mandatory, such situations cannot be resolved by the business logic.
5. **Other Errors.** Many relations are not recognized by the models. Often such errors occur in sentences with less grammatical structure and sentences that contain many special characters like '@', '#' or typical trigger phrases that do not belong to any relation. The example shows a *Disaster* that no model predicted.

The results indicate that current event or binary relation extraction approaches outperform MARE models on the task of binary relation extraction. However, as we weaken the structural requirements, MARE models become superior. The introduced MARE approaches allow the extraction of complex multi-attribute relations from plain text without an enumeration of all relation candidates. The limitations of our approaches, see Section 4.2, had no severe impact concerning the smart data corpus.

6 CONCLUSION

We introduced multi-attribute relation extraction and differentiated this definition from current terminology as n-ary relation extraction and event extraction. Our problem definition leads to simplified approaches for extracting relations with an arbitrary amount of attributes by avoiding the usage of candidate enumeration and the trigger concept.

MARE models are superior if the relations do not fit into binary or event schema. They avoid structural constraints and perform better than current state-of-the-art relation and event extraction approaches on the SmartData corpus.

We plan to involve the manual analysis results in the construction of improved MARE approaches in the future. Mostly we want to address the limitations the MARE approaches have and the incorporation of the relation-specific context.

REFERENCES

Aguilar, J., Beller, C., McNamee, P., Van Durme, B., Strassel, S., Song, Z., and Ellis, J. (2014). A Comparison of the Events and Relations Across ACE, ERE, TAC-KBP, and FrameNet Annotation Standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53, Baltimore, Maryland, USA. Association for Computational Linguistics.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv:2003.10555 [cs]*. arXiv: 2003.10555.

Consortium, L. D. (2005). Ace (automatic content extraction) english annotation guidelines for events. page 77.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. arXiv: 1810.04805.

Eberts, M. and Ulges, A. (2019). Span-based Joint Entity and Relation Extraction with Transformer Pretraining. *arXiv:1909.07755 [cs]*. arXiv: 1909.07755.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. S. (2017). Allennlp: A deep semantic natural language processing platform.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2010). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991 [cs]*. arXiv: 1508.01991.

Kim, J.-D., Pyysalo, S., Ohta, T., Bossy, R., Nguyen, N., and Tsujii, J. (2011a). Overview of BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA. Association for Computational Linguistics.

Kim, J.-D., Wang, Y., Takagi, T., and Yonezawa, A. (2011b). Overview of Genia event task in BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 7–15, Portland, Oregon, USA. Association for Computational Linguistics.

Lai, P.-T. and Lu, Z. (2021). BERT-GT: Cross-sentence *n* -ary relation extraction with BERT and graph transformer. *Bioinformatics*.

Li, C. and Tian, Y. (2020). Downstream Model Design of Pre-trained Language Model for Relation Extraction Task. *arXiv:2004.03786 [cs]*. arXiv: 2004.03786 version: 1.

Liu, Y., Li, A., Huang, J., Zheng, X., Wang, H., Han, W., and Wang, Z. (2019). Joint Extraction of Entities and Relations Based on Multi-label Classification. In *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pages 106–111.

Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization. *arXiv:1711.05101 [cs, math]*. arXiv: 1711.05101.

May, P. and Reißel, P. (2020). German electra uncased.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - ACL-IJCNLP '09*, volume 2, page 1003, Suntec, Singapore. Association for Computational Linguistics.

Mountassir, A., Benbrahim, H., and Berrada, I. (2012). An empirical study to address the problem of unbalanced data sets in sentiment classification. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3298–3303.

Peng, N., Poon, H., Quirk, C., Toutanova, K., and Yih, W.-t. (2017). Cross-Sentence N-ary Relation Extraction with Graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Roller, R., Rethmeier, N., Thomas, P., Hübner, M., Uszkoreit, H., Staeck, O., Budde, K., Halleck, F., and Schmidt, D. (2018). Detecting Named Entities and Relations in German Clinical Reports. In Rehm, G. and Declerck, T., editors, *Language Technologies for the Challenges of the Digital Age*, Lecture Notes in Computer Science, pages 146–154, Cham. Springer International Publishing.

Schiersch, M., Mironova, V., Schmitt, M., Thomas, P., Gabryszak, A., and Hennig, L. (2018). A German Corpus for Fine-Grained Named Entity Recognition and Relation Extraction of Traffic and Industry Events. page 8.

Tsujii, J., Kim, J.-D., and Pyysalo, S., editors (2011). *Proceedings of BioNLP Shared Task 2011 Workshop*, Portland, Oregon, USA. Association for Computational Linguistics.

Viera, A. J., Garrett, J. M., et al. (2005). Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363.

Wadden, D., Wennberg, U., Luan, Y., and Hajishirzi, H. (2019). Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Xiang, W. and Wang, B. (2019). A Survey of Event Extraction From Text. *IEEE Access*, 7:173111–173137. Conference Name: IEEE Access.

Xu, F., Uszkoreit, H., Li, H., Adolphs, P., and Cheng, X. (2013). Domain Adaptive Relation Extraction for Semantic Web.

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., and Xu, B. (2017). Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *arXiv:1706.05075 [cs]*. arXiv: 1706.05075.