Reliable Rules for Relation Extraction in a Multimodal Setting

Björn Engelmann[1] Philipp Schaer [2]

Abstract: Relation extraction for automated knowledge base construction typically requires much training data. If these are not available for a specific information need, relations must be extracted manually, or by hand-crafted extraction rules [Wu18]. Data Programming can be used to define heuristics that generate noisy labels for many instances, but this requires programming knowledge [Di19]. We present an approach to extract relations from multimodal documents using a few training data. Furthermore, we derive explanations in the form of extraction rules from the underlying model to ensure the reliability of the extraction. Finally, we will evaluate how reliable (high model fidelity) extracted rules are and which type of classifier is suitable in terms of F1 Score and explainability.

Keywords: Relation Extraction; Knowledge Extraction; Knowledge Base Construction; Explainable AI; Multimodal Documents

1 Introduction

Automatic knowledge base construction is a task that typically requires a large amount of labelled data against which extraction models can be trained. Unfortunately, especially in relation extraction, these labels are often unavailable since concrete use cases frequently differ strongly from each other. The corpus documents vary regarding the language used, data modality, structuredness, and domain. Many documents are also multimodal, which means that in addition to the text, they contain much other information, such as tables, font size, and text alignment. For this reason, annotated relations often must first be created in a laborious and error-prone procedure [Wu18].

An alternative to manual annotation is the application of hand-crafted extraction rules, which can automatically create noisy labels for many data instances [Ra17]. However, creating these rules requires programming knowledge, and without gold data, it is impossible to evaluate whether the applied rules are reliable. Furthermore, the application of complex language models in the low-resource setting is often infeasible due to the high computational effort involved [Ga21].

Due to these challenges, we present an approach that finds reliable extraction rules based on a small number of annotations, which makes the extraction model explainable on the

[1] Technische Hochschule Köln, Information Retrieval Research Group bjoern.engelmann@th-koeln.de
[2] Technische Hochschule Köln, Information Retrieval Research Group philipp.schaer@th-koeln.de

doi:10.18420/BTW2023-69

one hand and suitable to annotate new relations on the other hand. This has the advantage that a user can assess the model's reliability without having a lot of test data available. A small amount of training data is called a number less or equal to 10 instances in our context of information extraction. We follow the convention of few-shot learning, where few-shot means that only a few labelled training instances are available [De22]. These extraction rules are presented to a user, who can decide by expert feedback whether a rule fits their use case. The requirements for such a user are generally the necessary domain knowledge about the relations of the use case and basic HTML knowledge. A typical user group for this are data journalists, who often have HTML knowledge but not necessarily programming knowledge.

This work presents an approach that allows the integration of expert knowledge and expands the group of potential users for extracting relations in a low-resource setting without much labeling or programming effort. To ensure this, our model requires only a small amount of training data and provides extraction rules with high fidelity, which are suitable for user-driven feedback. For this purpose, we combine approaches from Explainable AI with those from Data Programming. If these rules are considered reliable by a user, it can reduce the annotation process for a new data science project.

The remainder of this paper is structured as follows: section 2 describes relevant related work. Then, in section 3, we introduce our overall approach using the associated pipeline and the methodological details. Next, our evaluation, experimental settings and the dataset used are presented in section 4. Finally, section 5 discusses the results and shows future work.

2 Related Work

The construction of a Knowledge Base is challenging, as Knowledge Bases need to be accurate, up-to-date, comprehensive, flexible, and efficient as possible. [Di19] propose automated knowledge base construction requirements that are not fully covered by any of the systems studied. An important key feature is an option for user feedback in which they can define or select extraction rules without coding skills.

Fonduer [Wu18] is a tool that implements a complete pipeline for the extraction of relations. It is based on the fact that users define the extraction rules themselves and evaluate and constantly improve them in an iterative process. Documents are automatically parsed, and their multimodal information, such as the membership of a span to a table header, is preserved. The user-defined extraction rules thus form entity types and relation types. However, these rules require programming knowledge, and the final extraction model can only be explained indirectly based on the defined rules.

By 2003, several methods for adaptive information extraction had already been presented. These focused roughly on two approaches. On the one hand, knowledge extraction with the

help of finite state techniques which are expressed by grammars or automata. On the other hand, relational rule learning techniques, where rules are learned in a Prolog style [KT03].

Modern approaches based on language models can consider the context of entities (neighbourhood of elements in the DOM tree) in HTML documents for extraction. In a few-shot setting, attributes can be extracted from a web page by pre-training the model on unlabeled web pages [De22]. In this way, an average of 10 training websites is sufficient to achieve an attribute value-level F1 score of 94.2 for an attribute extraction task on a website.

[Ha17] have presented a tool that provides a user with a visual interface to perform simple information extraction tasks without knowing a programming language. Easy to understand extraction rules are presented, which are generated from a small set of labelled data. Their system already has a bunch of predefined rules. These rules can then be refined to improve extraction performance. However, it is impossible to relate these rules to a trained model, which makes it impossible to perform more complex extraction tasks.

Explainable Artificial Intelligence is a research field that aims to make AI systems results more understandable to humans [AB18]. Approaches to make the behaviour of these black-box systems understandable are, e.g., Rule Extraction or Feature Importance. Lime [RSG16a] can be used to generate explanations in the form of feature importance scores within a local neighbourhood around the instance to be explained. To do this, new artificial instances are sampled near the input to be explained, which are then used to learn a simple local regression model. The learned coefficients then correspond to the feature scores. To evaluate the comprehensibility of extracted explanations, [Ji21] conducted a survey and assessed which types of explanations are well suited to improve the understanding of a black-box model. Users found extracted rules very helpful, especially when they refer to a few features.

To link user feedback with Explainable Artifical Intelligence, [TK18] give a user the possibility to mark a feature for a given training example in such a way that this feature should not influence the training process. The user can determine whether the model has mistakenly drawn a connection between this feature and the example through domain knowledge.

The advantage of our approach is that arbitrary extraction models can be used, from which explanations can be extracted using Explainable AI. This is not provided for in the classical techniques of adaptive learning. In principle, our approach is compatible with all black-box classifiers and all XAI methods that provide extraction rules. Prolog-like rule systems, for example, do not offer this flexibility since, at most, the rule systems themselves can serve as explanations.

To our knowledge, no system has been presented that allows users without programming knowledge to extract reliable relations from multimodal data based on a few training examples.

3 Methods

This section describes the individual building blocks of our approach. In Figure 1, the procedure is roughly shown.

A knowledge base construction framework (Fonduer [Wu18]) parses multimodal documents, and relation candidates are generated. The user labels a small set of documents (subsection 3.1).

Relations go through the steps of featurization, dimension reduction and feature combination (subsection 3.2, subsection 3.3).

The transformed vectors are used to train the model. The model provides predictions and feedback to the user through a ranked list of explanations (subsection 3.4).

The user gives feedback in the form of a selection of reliable rules (subsection 3.5).

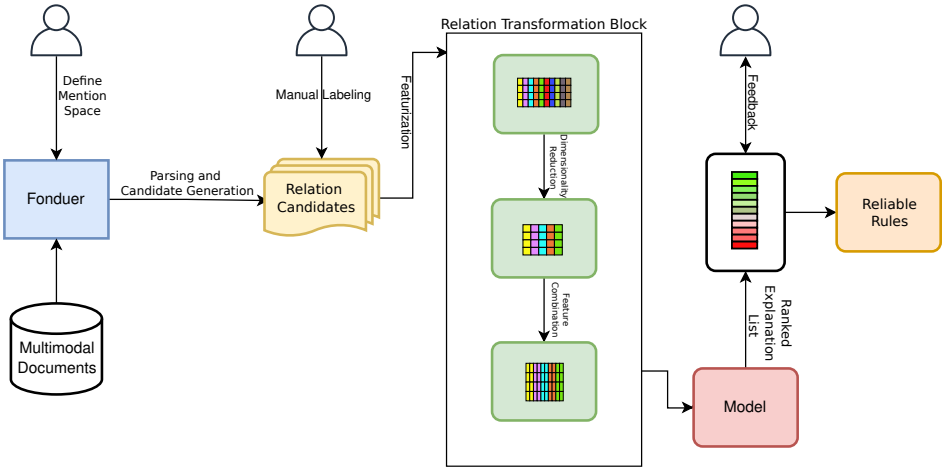

Fig. 1: Process of the overall approach in a productive environment. In the relation transformation block, rows represent instances, columns express individual entity features, and colours symbolise a component in the feature vector.

3.1 Parsing and Candidate Generation

Parsing multimodal documents is challenging because different kinds of information (font, table structure, colour) should be preserved. However, as much information as possible should be available in the database in a uniform and structured way. For this purpose, the framework Fonduer is used in this work [Wu18]. After parsing, a hierarchically structured graph is available for each document. For example, a section can contain text, tables, or

figures. The smallest unit consists of sentences. Fonduer captures the context of sentences and candidates, e.g. information about where they appear in the document. Furthermore, details such as font size or the HTML class used for an element are preserved.

This form of modelling makes it possible to use the structural information of the document as a signal for relation extraction. In our case, relation candidates consist of two mentions, which are two text spans in the document that potentially express the user's desired relation. Both correct and incorrect relation candidates are required to train the model to solve the binary classification problem. Correct relations are those candidates where both entity mentions are in a predefined connection. Incorrect relations contain entity mentions that are randomly drawn from the document. Accordingly, all relation candidates are derived from the cartesian product of both mention sets. Furthermore, we make sure that some incorrect candidates contain exactly one correct entity mention (details in subsection 4.2). The set of correct relation candidates is manually assigned. Fonduer then transforms these candidates into a feature space that embeds textual, structural, visual, and tabular features. We denote the transformed correct candidates as $\mathcal{R}_{pos}$ and the incorrect $\mathcal{R}_{neg}$, respectively.

3.2 Featurization and Dimensionality Reduction

With the Fonduer featurization, each relation candidate is assigned to a feature vector of dimension D derived from the multimodal context of the linked entities. Each feature is binary and expresses whether a property applies to a relation or not (e.g., the first entity mention contains the word *professional*, some instances can be seen in Table 2). This feature vector is denoted as $\mathbf{r} = \{r_1, ..., r_D\}$. Each component r_i of the feature vector refers to a property of the respective entity mention of entity type e_0 or e_1. We denote the set of all properties of an entity type $\mathcal{E}_0$ and $\mathcal{E}_1$, respectively. Furthermore, some properties refer to the context of both entity mentions (e.g., the distance of both mentions from each other), the total set of which we call $\mathcal{G}$. We denote the set of all features $\mathcal{F} = \mathcal{E}_0 \cup \mathcal{E}_1 \cup \mathcal{G}$.

The dimensionality of a feature vector is typically over 100k for a common set of HTML documents. However, since we have little training data available and want to make model decisions explainable, we reduce the dimensionality [Cu08]. This is achieved by filtering out the vector components that vary least from the difference between the average correct and incorrect training vectors. We use this form of dimension reduction to obtain binary features. Since the explainability of single features should be preserved, we cannot use techniques like singular value decomposition or embeddings because we would obtain a real feature space. Our dimension reduction approach assumes that those features are particularly relevant for the classification whose values differ most between the positive and negative candidates. The l most varying components are defined in the following way:

$$\mathcal{F}_{sig} = \text{argsort}_l \left(|\overline{\mathbf{r}_{pos}} - \overline{\mathbf{r}_{neg}}| \right) . \tag{1}$$

$$\overline{\mathbf{r}_{pos}} = \frac{1}{|\mathcal{R}_{pos}|} \sum_{\mathbf{r}^{(i)} \in \mathcal{R}_{pos}} \mathbf{r}^{(i)}. \tag{2}$$

The argsort$_l$ function returns indices of the l vector components with the highest value. The average of all vectors corresponding to the incorrect relations $\overline{\mathbf{r}_{neg}}$ is defined respectively. Thus, $\mathcal{F}_{sig}$ contains the l indices of such components that differ most with respect to correct and incorrect relations. We assume that the corresponding features are the most important for our relation classification task and discard the rest.

3.3 Feature Combination

Another approach we take is to combine features from $\mathcal{E}_0$ and $\mathcal{E}_1$ into one feature. As explained in subsection 3.1, there are incorrect relations where one of two entity mentions is correct. This is because each relation, in our case, consists of exactly two entities. Since individual features are intended to serve as both a classification explanation and an extraction rule, it is reasonable to require the validity of two relation properties in one rule. Therefore, explanations and extraction rules should also express both properties in one. Here we use only those features that are preserved after dimension reduction. We denote the set of all combinations of $\mathcal{E}_0$ and $\mathcal{E}_1$ together with the features from $\mathcal{G}$, $\mathcal{F}_{combined}$, where $\mathcal{F}_{combined} = (\mathcal{E}_0 \times \mathcal{E}_1) \cup \mathcal{G}$.

3.4 Training and Explanations

For the binary classification task, we use low-complexity models (details in subsection 4.2) because their training is better suited in the low-resource setting, and there is evidence that the complexity of models correlates negatively with their explainability [Gu19].

After the training, we extract a set of explanations with Lime. Lime is an approach that explains the prediction of a specific instance based on the importance weights of the associated features [RSG16b]. Lime explains a selected instance, but since we want to obtain extraction rules that explain the overall model and provide valuable explanations, we need to choose a representative but also a diverse set of instances for Lime. We have chosen this local approach because the dimension reduction already performs a global selection, and the local approach results in a multitude of explanations. In addition, this has the advantage that a user can select one of these explanations. We generate diversified artificial instances based on our test relations to obtain explanations that cover as many relation patterns as possible. We build clusters over our test vectors using the k-means procedure to achieve this. The rounded cluster centres then form our representative, diverse instances based on which explanations are extracted. We derive a ranked list of feature combinations by summarizing the weights over all instances and sorting them in descending order. The intuition is that

each feature can be interpreted as a rule to classify unseen data. The higher the explanation weight of a rule is, the closer its predictions are to the predictions of the explained model.

Another approach to measuring the fidelity of each feature to the model is to apply each feature, interpreted as a rule, to the test data and then compare the results to the model predictions. Thus, a baseline is established that assigns an F1 Fidelity to each rule, which is derived from the F1 Score of the model predictions and the application of each rule [Gu18a].

3.5 User Feedback

The ranked explanation list can then be presented to a user who selects a reliable rule based on domain knowledge to classify relation candidates. Under the assumption that the list position correlates with the actual F1 Score, the advantage is that a user has less effort in selecting a reliable rule. In subsection 4.4, an example of a ranked list is shown, in addition to the quantitative analysis, to make plausible that some rules are both understandable and accurate. We present rules that we assume a user would plausibly choose to find an appropriate expression based on the context of the use case.

4 Experiments

The following subsections evaluate which classifiers are suitable for relation candidate classification and generating reliable extraction rules based on different amounts of training data. The test data labels are only used to evaluate the final results. We never use the test labels for dimension reduction or hyperparameter selection. Our code and data are available at `https://osf.io/dn9hm/?view_only=7e65fd1d4aae44e1802bb5ddd3465e08`.

4.1 Dataset

The Structured Web Data Extraction (SWDE) dataset consists of a collection of 124,291 structured web pages with 8 different verticals [Ha11]. A vertical (e.g., job posting) consists of 10 differently formatted websites, each consisting of up to 2000 pages. Within a page are labelled attributes (in the case of job postings: title, company, location, date). For our case, we want to extract relations between job titles and corresponding locations. All websites are available in HTML. Since there can be many mentions of job titles and locations on each website, only those relation mentions are considered correct whose entity mentions are at the correct position in the document. An example of a job posting can be seen in Figure 2. We use 400 webpages from the Careerbuilder site. We define the mention space for all jobtitle entity mentions as n-grams between 1-9 items and 1-6 items for the location mentions, respectively.

Fig. 2: Example excerpt of a job posting from the Careerbuilder website. The entities of the correct relation are marked in green.

4.2 Experimental Settings

For the experiments, different types of models are used to investigate the relationship between classification performance and explainability. For all of the following models, the sklearn default configurations are used: Multi-Layer Perceptron (MLP), Decision Tree (DT), k-Nearest Neighbors (kNN), Gradient Boosting (GB), Random Forest (RF), Support Vector Classifier (SVM), Naive Bayes (NB) [Pe11]. Since we want to evaluate our approach for a small number of training data, we limit the number of correct relations $|\mathcal{R}_{pos}|$ and test the following amounts: $|\mathcal{R}_{pos}| \in \{3, 5, 10, 20, 40\}$. We sample 10 incorrect relations for each correct relation, which is a typical ratio for relation extraction [NG15]. Under these 10 incorrect relations are two containing exactly one correct entity span. Since we know that a document can contain only one correct relation, the remaining combinations of mentions can serve as the basis of the incorrect relations. We use this variety of simple classifiers to evaluate whether differences in explainability can be detected. We also use the standard deviation of classification performance over multiple training runs to assess how reliable a model is for a given set of training data. The larger the standard deviation of the F1 score of a model, the less reliable the extraction performance.

4.3 Ablations

We evaluate each module of our pipeline in terms of median F1 Score. Thus, each model type is evaluated with the totality of all features, with the features after dimensionality reduction, and with the combined features. Each configuration is evaluated with 10-Fold cross-validation to determine standard deviation and median values. The scatter values of the classifiers are particularly important since this is an indicator of model reliability. We use the scatter of the F1 Score to measure the model's reliability, as we don't know the actual F1 Score in a productive setting where we have no labelled test data. Especially when little training data is used, a higher scatter of F1 Score results (as seen in Table 1).

For k-means clustering, we use 10 cluster centres, and dimension reduction reduces the feature space from 382k dimensions to 30. Dimension reduction almost always resulted in better F1 Scores. This can be seen particularly clearly for Naive Bayes and SVM. When applying the feature combination, no clear pattern emerges; only the standard deviation for the Random Forest decreases and a constantly increased F1 Score for Naive Bayes. It is also noticeable that for the Random Forest model with combined features in the median already, 5 correct training relations are sufficient to achieve an F1 Score of 1.0, with a standard deviation of 0.1.

Tab. 1: Median F1 Scores and corresponding standard deviations for different training amounts and model types.

Model / # train	2	3	5	10	20	40
MLP full	0.84±0.06	0.85±0.03	0.87±0.01	0.9±0.02	0.94±0.02	0.95±0.02
MLP red.	0.96±0.08	0.98±0.06	0.99±0.07	1.0±0.02	1.0±0.01	1.0±0.0
MLP comb.	0.96±0.04	0.97±0.02	0.98±0.02	0.99±0.01	1.0±0.01	1.0±0.0
DT full	0.86±0.12	0.79±0.1	0.85±0.06	0.9±0.04	0.94±0.03	0.95±0.01
DT red.	0.86±0.12	0.85±0.06	0.92±0.08	0.94±0.05	0.96±0.03	1.0±0.01
DT comb.	0.93±0.05	0.91±0.07	0.97±0.03	1.0±0.04	1.0±0.01	1.0±0.0
KNN full	0.2±0.32	0.29±0.15	0.42±0.12	0.57±0.11	0.63±0.07	0.72±0.07
KNN red.	0.97±0.01	0.98±0.02	0.99±0.01	1.0±0.01	1.0±0.01	1.0±0.0
KNN comb.	0.97±0.06	0.98±0.01	0.99±0.01	1.0±0.01	1.0±0.01	1.0±0.01
RF full	0.93±0.27	0.97±0.04	0.94±0.03	1.0±0.02	1.0±0.01	1.0±0.0
RF red.	0.93±0.09	0.96±0.09	0.98±0.08	1.0±0.01	1.0±0.01	1.0±0.0
RF comb.	0.98±0.01	0.98±0.02	**1.0±0.01**	1.0±0.01	1.0±0.01	1.0±0.0
GB full	0.86±0.04	0.85±0.06	0.93±0.07	0.94±0.05	0.96±0.04	0.97±0.01
GB red.	0.86±0.06	0.85±0.07	0.95±0.08	0.96±0.05	0.95±0.03	1.0±0.01
GB comb.	0.97±0.01	0.98±0.02	0.98±0.01	0.99±0.01	1.0±0.02	1.0±0.0
NB full	0.01±0.24	0.07±0.27	0.64±0.26	0.64±0.3	0.96±0.02	0.92±0.03
NB red.	0.74±0.17	0.76±0.12	0.75±0.02	0.78±0.06	0.77±0.04	0.77±0.01
NB comb.	0.9±0.27	0.89±0.06	0.94±0.03	0.92±0.03	0.93±0.03	0.96±0.02
SVM full	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
SVM red.	0.93±0.17	0.96±0.03	0.97±0.03	1.0±0.03	1.0±0.02	1.0±0.0
SVM comb.	0.87±0.17	0.9±0.07	0.92±0.07	0.98±0.05	0.99±0.02	1.0±0.0

4.4 Explanation Evaluation

Tab. 2: Selection of the 10 extracted rules with the highest F1 Fidelity.

F1 Fidelity Explanation	F1 Score
STR_e0_HTML_ATTR_class=job_title AND STR_e1_NEXT_SIB_TAG_iframe	1.0
STR_e0_HTML_ATTR_class=job_title AND BASIC_e1_CONTAINS_WORDS_[US]	1.0
STR_e0_HTML_ATTR_class=job_title AND STR_e1_HTML_ATTR_rel=nofollow	1.0
STR_e0_HTML_ATTR_class=job_title AND BASIC_e1_CONTAINS_WORDS_[US -]	1.0
STR_e0_HTML_ATTR_class=job_title AND STR_e1_HTML_ATTR_class=BingMap	1.0
STR_e0_HTML_ATTR_class=job_title AND STR_e1_HTML_ATTR_id=JobDetails_..	1.0
STR_e0_ANCESTOR_TAG_[html body ...] AND STR_e1_NEXT_SIB_TAG_iframe']	0.98
STR_e0_ANCESTOR_TAG_[html body ...] AND BASIC_e1_CONTAINS_WORDS_[US]	0.98
STR_e0_ANCESTOR_TAG_[html body ...] AND STR_e1_HTML_ATTR_rel=nofollow'	0.98
STR_e0_ANCESTOR_TAG_[html body ...] AND STR_e1_HTML_ATTR_class=BingMap	0.98

Since our goal is to extract reliable rules, we evaluate the fidelity of the explanations of all model types using the F1 Fidelity between explanation and prediction [Gu18b]. To evaluate the quality of the final ranking, we calculate the rank correlation between an optimal ranked list (according to the F1 Score for a specific rule against the test labels) and a list resulting from ordering the explanation weight. The set of rules to be ordered is the same here, only the order may differ. We used the Spearman rank correlation instead of the Pearson correlation coefficient since the values of the explanation weights are no longer relevant for the ranking, only their order. Furthermore, the distribution of the explanation weights does not necessarily follow a normal distribution, which must be assumed for the Pearson correlation coefficient.

Figure 3 illustrates how the number of training data, the model, and the explanation type affect the rank correlation. Extracted rules ordered by F1 Fidelity correlate more strongly with an optimally ranked list than a list ordered by Lime explanation weights. Furthermore, it is shown that RF and KNN achieve a rank correlation of more than 0.98 from 5 correct training relations. The Lime explanation weights for SVM and DT were omitted because they do not have a function to assign pseudo-probabilities to instances. In general, the rank correlation tends to improve for an increasing number of training data.

In Table 2, the top ten explanations extracted from a Gradient Boost model are shown as an

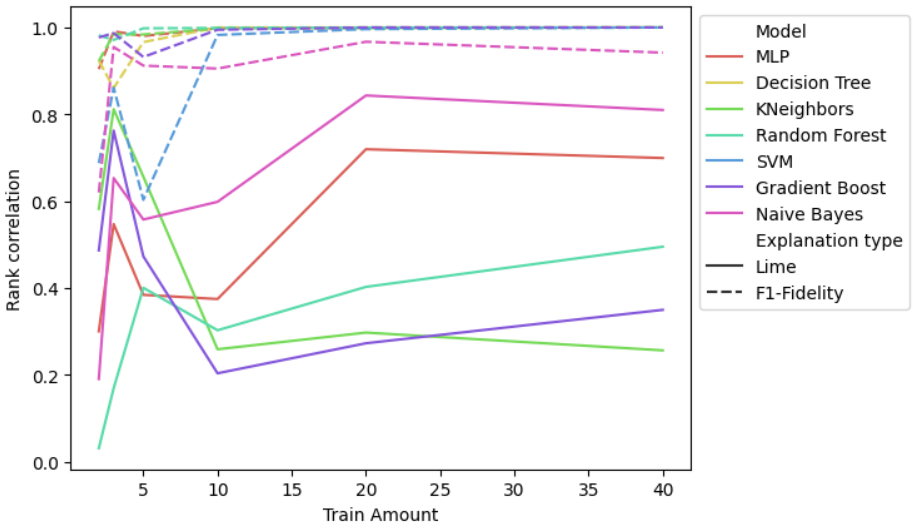

Fig. 3: Rank correlations for different models and explanation types plotted against the number of correct training relations. Only for models with combined features.

example. This was trained with 5 correct relations. According to Table 1, this configuration has an F1 Score of 0.98, while all top six extracted rules have an F1 Score of 1.0. We assume that a user would select rule #5 as reliable. Based on the HTML classes, the user can infer the meaning of the entities because the *jobtitle class* indicates a correct jobtitle mention, and the *BingMap class* expresses the presence of a corresponding location (Figure 2). The authors from [De22] use a complex language model to achieve an attribute value-level F1 score of 94.8 for a similar task. The results cannot be compared directly because their model does not use candidate generation; therefore, it is not a classification task but an attribute extraction task. Also, the model is trained on 80 different sites and thus has to recognize a larger variety of patterns. However, an average number of 10 webpages was used for the few-shot training.

5 Discussion

In this work, we presented an approach to extract relations and corresponding rules from multimodal documents using a small amount of training data. Using our example from Table 2, it can be seen that even a single rule can provide better extraction performance than the underlying model. The prerequisite for this is that a user would select this rule.

In this way, annotating new websites with less labelling effort is possible. This is the case because the user would have to use part of the annotated data in a setting without explanations to evaluate the extraction model. However, more than 5 annotated websites

would be necessary for a reliable evaluation. Reliable rules can then be used to annotate unknown data.

Rules extracted by Lime perform worse than those extracted by the baseline method. We assume this is because Lime is unsuitable for classification problems where many features provide a strong signal for the correct class. The main area for improvement in this work is the simplicity of the data set and the associated extraction task. Future work is to apply the presented approach to more complex data. Furthermore, more advanced approaches to explanatory extraction, such as Lore [Gu18c], will be used.

Bibliography

[AB18] Adadi, Amina; Berrada, Mohammed: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6:52138–52160, 2018.

[Cu08] Cunningham, Pádraig: Dimension Reduction. In (Cord, Matthieu; Cunningham, Pádraig, eds): Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 91–112, 2008.

[De22] Deng, Xiang; Shiralkar, Prashant; Lockard, Colin; Huang, Binxuan; Sun, Huan: , DOM-LM: Learning Generalizable Representations for HTML Documents, 2022.

[Di19] Din, Osman: Towards a Flexible System Architecture for Automated Knowledge Base Construction Frameworks. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 3066–3071, 2019.

[Ga21] Ganesh, Prakhar; Chen, Yao; Lou, Xin; Khan, Mohammad Ali; Yang, Yin; Sajjad, Hassan; Nakov, Preslav; Chen, Deming; Winslett, Marianne: Compressing Large-Scale Transformer-Based Models: A Case Study on BERT. Transactions of the Association for Computational Linguistics, 9:1061–1080, 09 2021.

[Gu18a] Guidotti, Riccardo; Monreale, Anna; Ruggieri, Salvatore; Pedreschi, Dino; Turini, Franco; Giannotti, Fosca: , Local Rule-Based Explanations of Black Box Decision Systems, 2018.

[Gu18b] Guidotti, Riccardo; Monreale, Anna; Ruggieri, Salvatore; Pedreschi, Dino; Turini, Franco; Giannotti, Fosca: , Local Rule-Based Explanations of Black Box Decision Systems, 2018.

[Gu18c] Guidotti, Riccardo; Monreale, Anna; Ruggieri, Salvatore; Pedreschi, Dino; Turini, Franco; Giannotti, Fosca: Local Rule-Based Explanations of Black Box Decision Systems. CoRR, abs/1805.10820, 2018.

[Gu19] Gunning, David; Stefik, Mark; Choi, Jaesik; Miller, Timothy; Stumpf, Simone; Yang, Guang-Zhong: XAI—Explainable artificial intelligence. Science Robotics, 4(37):eaay7120, 2019.

[Ha11] Hao, Qiang: Structured Web Data Extraction Dataset (SWDE). 2011.

[Ha17] Hanafi, Maeda F.; Abouzied, Azza; Chiticariu, Laura; Li, Yunyao: SEER: Auto-Generating Information Extraction Rules from User-Specified Examples. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, p. 6672–6682, 2017.

[Ji21] Jin, Weina; Fan, Jianyu; Gromala, Diane; Pasquier, Philippe; Hamarneh, Ghassan: EUCA: A Practical Prototyping Framework towards End-User-Centered Explainable Artificial Intelligence. CoRR, abs/2102.02437, 2021.

[KT03] Kushmerick, Nicholas; Thomas, Bernd: Adaptive Information Extraction: Core Technologies for Information Agents. In (Klusch, Matthias; Bergamaschi, Sonia; Edwards, Pete; Petta, Paolo, eds): Intelligent Information Agents: The AgentLink Perspective. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 79–103, 2003.

[NG15] Nguyen, Thien Huu; Grishman, Ralph: Relation extraction: Perspective from convolutional neural networks. In: Proceedings of the 1st workshop on vector space modeling for natural language processing. pp. 39–48, 2015.

[Pe11] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[Ra17] Ratner, Alexander; Bach, Stephen H.; Ehrenberg, Henry R.; Fries, Jason Alan; Wu, Sen; Ré, Christopher: Snorkel: Rapid Training Data Creation with Weak Supervision. CoRR, abs/1711.10160, 2017.

[RSG16a] Ribeiro, Marco Túlio; Singh, Sameer; Guestrin, Carlos: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. CoRR, abs/1602.04938, 2016.

[RSG16b] Ribeiro, Marco Tulio; Singh, Sameer; Guestrin, Carlos: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144, 2016.

[TK18] Teso, Stefano; Kersting, Kristian: , "Why Should I Trust Interactive Learners?Ëxplaining Interactive Queries of Classifiers to Users, 2018.

[Wu18] Wu, Sen; Hsiao, Luke; Cheng, Xiao; Hancock, Braden; Rekatsinas, Theodoros; Levis, Philip; Ré, Christopher: Fonduer: Knowledge Base Construction from Richly Formatted Data. In: Proceedings of the 2018 International Conference on Management of Data. SIGMOD '18, Association for Computing Machinery, New York, NY, USA, p. 1301–1316, 2018.