



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Master Thesis

Transformer-encoder-based financial entity and value extraction with distant supervision

Fabian Rausch

fabian.rausch@studium.uni-hamburg.de

Universität Hamburg

MIN Faculty - Department of Informatics - Language Technology Group

Course of Studies: M. Sc. Information Systems

Matriculation Number: 7325286

First reviewer: Prof. Dr. Chris Biemann

Second reviewer: Dr. Seid Muhie Yimam

Supervisor: M. Sc. Steffen Remus

Submission date: 29.06.2022

Abstract

Financial statements contain information about financial entities and associated financial values in unstructured, natural language form. Machine extraction of entities and relations from text is challenging due to the high flexibility of language and has most recently been addressed with AI models that learn and generalize with human annotated training examples.

The research questions of this thesis are whether a training dataset of annotated financial entities and values can be generated by machine without human assistance, and whether an AI model can learn and generalize with this training dataset so that financial entities and values can be automatically extracted from natural language text. The hypothesis is that a training dataset can be generated using financial reports, for each of which the unstructured natural language form and a structured key-value dictionary form are available. The structured representation serves for distant supervision learning. The resulting dataset can be used to learn a general entity detection and relation extraction model, which is adapted to the problem domain and its individual properties.

To investigate the research questions, a state-of-the-art joint entity detection and relation extraction model is analyzed and extended by encoding explicit linguistic information, specifically part-of-speech (POS) tags and dependency (DEP) tags. The original model is based purely on bidirectional transformer encoder representations (BERT) and the influence of the parsing information is evaluated on a public dataset. A method to algorithmically create a training dataset with financial entities and values using distant supervision is presented. For this purpose, the entities and values from the structured form are searched and annotated in the natural language text using string matching approaches. The resulting dataset is augmented in order to increase the diversity of the data and multiply particularly high quality training examples. Finally, it is used to train the extended joint entity detection and relation extraction model to extract financial entities and their values from natural language text.

The evaluation is performed empirically using human annotated test data. For the joint extraction of financial entities and values, an F1 score of 81.55 is achieved with a manually created training dataset. For the distant supervision dataset, this score is significantly lower at 67.59. The use of POS and DEP tags in the model, data augmentation and the use of a fine-tuned BERT model all prove to be helpful measures on the test dataset.

Contents

1. Introduction	1
1.1. Problem context and motivation	1
1.2. Research questions	3
1.3. Hypothesis	3
1.4. Methodical approach	3
1.5. Structure of this work	4
2. Background	5
2.1. Annual financial statements	5
2.1.1. Statement components	5
2.1.2. Statement disclosure	5
2.2. XBRL	6
2.2.1. XBRL instance document and taxonomy	6
2.2.2. Standard taxonomies	8
2.3. Federal Gazette financial statement publications	8
2.3.1. Specific presentation from the Federal Gazette	9
2.3.2. Official publication files	9
2.3.3. Protection against data crawling	9
2.3.4. Validatis data purchase	10
2.4. String matching	10
2.4.1. Set-based similarity	11
2.4.2. Sequence-based similarity	11
2.4.3. Word embedding similarity	12
2.5. Linguistic parsing	13
2.5.1. Part-of-speech parser	13
2.5.2. Dependency parser	13
2.6. Entity detection	14
2.7. Relation extraction	14
2.8. Distant supervision	15
2.9. BERT	15
3. Related work	19
3.1. Entity detection and relation extraction	19
3.2. Financial statement information extraction	21

3.3. Research gap	23
4. Joint entity detection and relation extraction approach	25
4.1. Span-based entity and relation transformer	25
4.2. Model extension	28
4.2.1. Part-of-speech tags	28
4.2.2. Dependency tags	29
4.2.3. Shortest dependency path	30
4.3. Evaluation	31
5. Distant supervision for financial report annotation	33
5.1. Dataset	33
5.1.1. XBRL instance documents	33
5.1.2. XBRL taxonomy	36
5.1.3. XBRL label file	36
5.1.4. Text data	38
5.1.5. XBRL taxonomy references	39
5.1.6. XBRL financial entities	40
5.2. Approach	40
5.2.1. Financial value annotation	41
5.2.2. Financial entity annotation	45
5.2.3. Improvement and completion approaches	48
5.3. Evaluation	56
5.4. Distant supervision algorithm output examples and analysis	58
5.5. Dataset creation with distant supervision	60
5.5.1. Annotated dataset statistics and properties	60
5.5.2. Data augmentation	61
6. Financial entity detection and relation extraction	65
6.1. Datasets	65
6.1.1. Automatically annotated training dataset	65
6.1.2. Automatically annotated and augmented training dataset	65
6.1.3. Manually annotated training dataset	66
6.1.4. Test dataset	66
6.2. Approach	66
6.3. Evaluation	67
7. Discussion	69
8. Conclusion	73
Bibliography	75

Affidavit	81
A. Federal gazette publications	83
A.1. Specific presentation from the Federal Gazette example	83
A.2. Official publication file example	84
A.3. Official publication directory overview example	85
B. Raw data basis examples	87
B.1. XBRL data	87
C. Dataset statistics and properties	95
C.1. XBRL entities	95
C.2. XBRL entities not resolvable	107
D. Algorithmic annotation	109
D.1. Performance evaluation	109

1. Introduction

1.1. Problem context and motivation

Financial statements convey information about the course of a business period of an organization/corporation. It contains the annual financial statements with its balance sheet and income statement as well as the management report and notes. Depending on the legal form, size of the company and applicable law, publication of the financial statement is mandatory for companies based in Germany. The publication documents in that case are published in the German Federal Gazette and are available to the public.

The financial statement documents contain unstructured data in the form of natural text as well as figures and tables. Tables are particularly suitable for presenting the balance sheet and income statement, while natural text takes up the numerical values and supplements them with explanatory notes in the management report and notes.

To counteract the unstructured data nature of financial statements, the eXtensible Business Reporting Language (XBRL), an XML-based financial statement reporting format, has been promoted internationally and is mandatory in Germany since 2012 besides the unstructured, textual report. XBRL documents provide financial and non-financial values using a key value dictionary with a given set of applicable keys based on requirements derived from reporting standards.

Figure 1.1 illustrates on the left side how both natural text and the XBRL key value dictionary carry the same information for a financial statement extract. Both presentations will result in the same knowledge graph for the company shown on the right side of the figure (incomplete for presentation purposes). Using the key value dictionary it is no great challenge to derive the graph structure algorithmically because the data is structured. Financial entities are stored using dot-separated notation to stem the hierarchy and the only difficulty is to decide which key is an entity and which is meta information on the company. That can be easily achieved if the set of keys is pre-defined, which is the case in XBRL. Deriving the knowledge graph algorithmically from the natural text however is very difficult due to the flexible nature of natural text. There are countless possibilities to transport the given information in this example using natural language.

As part of a company's annual audit, it must be ensured that the integrity between

Financial natural text

Fixed assets for fiscal year 2021 amount to EUR 1000, which corresponds to an increase of EUR 100 compared to the previous year. Of this amount, EUR 400 relates to intangible assets (previous year: EUR 350). Property, plant and equipment remain unchanged at EUR 300.

Financial key value dictionary

company name: Example Company Inc.
 unit: EUR
 assets: 1000
 assets.intangible: 400
 assets.prop_plant_equipment: 300

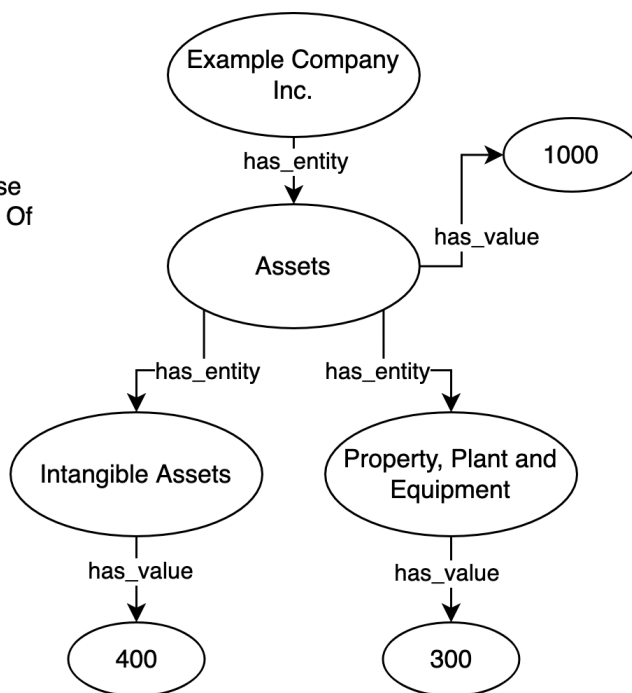


Figure 1.1.: Structured and unstructured nature of financial statements

the XBRL document and the natural language report, both part of the complete financial statement, is guaranteed. There must be no contradictions between these two documents. Comparing all numerical values between the two forms of presentation creates a large amount of manual work for auditors. Ideally, each firm would use a single, reliable data source and populate both the data in the XBRL documents and the text documents using variables from it so that no inconsistencies can arise. However, the auditor cannot rely on this and the digitization and use of professional software for this is not standard practice in companies. The manual maintenance of two data fields for the same entity is the resulting problem.

From the auditor's point of view, it follows that an algorithmic extraction of financial entities, their values and relationships among each other from natural text would offer a great reduction in workload. A knowledge graph could be automatically created from the extracted data and compared with the knowledge graph created from the XBRL document. Inconsistencies could be detected automatically. On top of that the graph representation allows for intrinsic consistency checks. In this example it could be checked whether all sub-entities of assets actually sum up to the value 1000. Finally, a generalized approach would allow to extract paired financial entity / value information from financial statements even if no structured XBRL files are available. The aim of this work therefore is to automatically extract financial entities and the corresponding values from text in order to create a key value dictionary and map the entities to an XBRL entity identifier to make it possible to create a graph from it.

1.2. Research questions

- Is it possible to annotate financial entities and financial values in the natural text of financial statement reports using the associated structured XBRL key value file in order to create a dataset algorithmically with distant supervision?
- Using the dataset created in the context of the first research question, is it possible to learn a generalized model that recognizes financial entities and their associated values from natural text in order to extract meaningful, structured key-value information from financial statement documents even without an XBRL file for distant supervision present?

1.3. Hypothesis

The structured XBRL representation and the natural text each refer to the same section of reality. 1.) The financial values of the XBRL key-value dictionary can be discovered in the text document via a string comparison of the sequence of digits. In the same sentence, the associated financial entity can now be searched for, since there is a plain text identifier for each XBRL entity provided by the linked XBRL taxonomy. Thus, pairs of financial value and financial entity can be automatically annotated in the text with the help of the XBRL file. 2.) With a sufficiently large amount of data annotated in this way, a model can then be learned that generalizes statistical, contextual linguistic features of financial values and financial entities within text. Finally, with this model, entities and values that were not part of the annotated training data can be also recognized and thus pairs of financial entities and financial values can be extracted from text independently from the XBRL file. Linking these entities to an XBRL entity of the taxonomy then allows the creation of a graph as shown in Figure 1.1.

1.4. Methodical approach

The motivation of the work and the hypothesis belonging to the research questions lead to the fact that this work focuses on prescriptive research with the aim of constructing an artifact in the form of algorithms and models that contribute to the solution of the described problem. This is the typical manifestation of prescriptive research (March and Smith, 1995). Accordingly, the research activities in this thesis will be the design, construction and evaluation of such an artifact. For these steps, particular attention will be paid to related work in order to build on their findings and to reveal the research gap. This is also provided for by the research guidelines introduced by Hevner et al. (2004) and the publication scheme for design-oriented research articles by Gregor and Hevner (2013), which thus give methodological research confidence to this approach.

The evaluation of the artifact is done with empirical-quantitative experiments, an appropriate method for design-oriented information systems research (Wilde and Hess, 2006) and for this specific case. Evaluation is essential for rigorous design-oriented research and for assessing the constructed artifact (Venable et al., 2012). Human annotated data is used for evaluation in this work. Thus, evaluation data can be used to measure whether the developed artifacts fulfill their intended utility under real-world conditions (Riege et al., 2009). For this purpose, the predictions of the models and algorithms are compared with the evaluation data that are accepted as correct. This results in a special limitation for the use of the research results, which is, however, typical for empirical evaluations. The performance results measured against the evaluation data cannot be generalized to the whole problem domain (Popper, 2005, p. 4). Accordingly, the research questions can only be answered in relation to the evaluation data, not in a general way. However, the evaluation data are taken at random from the entire data set and there is no particular reason for now to assume that these data are not representative of the entire domain of German financial statements data.

1.5. Structure of this work

Following this introduction, the underlying theoretical background of this thesis is presented in chapter 2. This includes e.g. the XBRL format and the basics of entity detection and relation extraction. This is followed by a literature review of the current state of research in the area of joint entity detection and relation extraction and a classification of this thesis in the research gap in chapter 3. After a general-purpose model for entity and relation extraction is developed in chapter 4, an approach for automated generation of a financial dataset using distant supervision is presented in chapter 5, answering the first research question. Chapter 6 finally connects the entity detection and relation extraction approach with the generated dataset and thus addresses the second research question. The thesis ends with a discussion of the final results and a conclusion.

2. Background

This chapter introduces the main theoretical background information forming the underlying foundation of this work.

2.1. Annual financial statements

At the beginning of a business and at the end of each financial year, a merchant/corporation shall prepare financial statements showing the relationship between assets and liabilities (section 242 HGB). The required components of the annual financial statements and the form in which they are to be submitted are set out for Germany in the German Commercial Code.

2.1.1. Statement components

According to legal requirements for corporations in Germany, financial statements have to contain the balance sheet (section 266 (1) sentence 2 HGB), income statement (section 275 HGB), notes (sections 284, 285 HGB), management report (sections 289, 289a HGB), signature (section 245 HGB), auditor's opinion/rejection (section 328 (1a) sentence 2 HGB), auditor's name (section 322 (7) sentence 1 HGB) and the information on the approval of the annual financial statements (section 328 HGB). Depending on the size of the corporation, slightly different requirements apply to the components of the annual financial statements (section 267 HGB). For partnerships, the annual financial statements consist only of the balance sheet and income statement (section 242 HGB).

2.1.2. Statement disclosure

Financial record keeping (section 238 HGB) and creating financial statements annually is mandatory for all merchants in Germany (section 242 HGB), (section 264 HGB) while exceptions due to certain threshold values are defined in section 241a HGB. The financial statement documents must be submitted in electronic form to the operator of the Federal Gazette¹ (Bundesanzeiger) for publication (section 264 (1) HGB)(section 325 (1) HGB) if the organization is a capital company (section 325 HGB) or a partnership exceeding certain thresholds (section 1 PublG). The deadline for each report is one year after the reporting date (section 325 (1a) HGB).

¹<https://www.bundesanzeiger.de/>

The Act on the Electronic Transmission of Balance Sheets and Profit and Loss Accounts forces companies to use an officially prescribed data format for electronic transmission to the Federal Gazette (section 5b EstG). This law stems from the Act to Modernize and Reduce Bureaucracy in the Tax Procedure (Tax Bureaucracy Reduction Act) and provides for various steps towards electronic and standardized reporting (BGBl. I 2008 p. 2850). With the letter from the Federal Ministry of Finance of January 2010, the *eXtensible Business Reporting Language* (XBRL) format therefore was defined as the technical transmission standard for reports from later than 2011 (BStBl. I 2010 p. 47) and the prescribed data format for report disclosure is the respective current XBRL taxonomy from the Federal Ministry of Finance (Bundesministerium für Finanzen, 2021), (Bundesministerium der Finanzen, 2018).

2.2. XBRL

The eXtensible Business Reporting Language (XBRL)² is an XML-based open-source format for the structured storage and transmission of corporate data (financial as well as non-financial). It is used internationally for the standardization of annual financial statements (Bundesministerium der Finanzen, 2018) and, as mentioned, is also the mandatory publication standard in Germany. XBRL is a construction kit with a set of abstract definitions of report elements that are semantically defined using a taxonomy. Assigning values to all taxonomy elements applicable to an enterprise forms a structured and complete financial report and is called XBRL instance document (XBRL Deutschland e. V., 2021a). In the context of this work, every financial concept defined in a XBRL taxonomy is called *financial entity* and every corresponding value is called *financial value*. Examples for such financial concepts are *assets*, *liabilities* etc.

2.2.1. XBRL instance document and taxonomy

Figure 2.1 visualizes the concepts of the XBRL instance document, the taxonomy, and their relationship to each other.

An XBRL instance document contains a set of facts belonging to the organization. Each of these facts refers on the one hand to a concept defined in the taxonomy using a tag id and on the other hand to further attributes such as the currency unit or the temporal context of the variable. Additionally, a value is assigned to each fact. Listing 2.1 shows an example for a single fact in an XBRL instance document.

Listing 2.1: XBRL fact example

```
1 <taxonomy:liab temporalContext="2021-12-31" unit="EUR">1000</
   taxonomy:liab>
```

²<https://de.xbrl.org/>

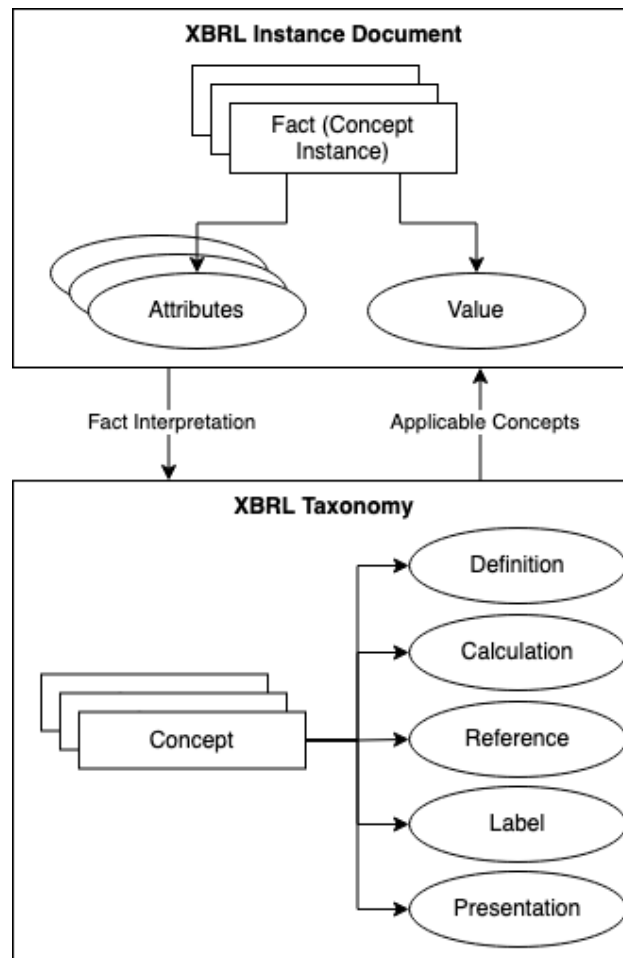


Figure 2.1.: XBRL instance document and XBRL taxonomy

In the example, the XML tag is first used to refer to a taxonomy concept (in this case *taxonomy:liab*, which is short for liabilities). The attributes for the temporal context and for the currency unit are then assigned values before the value of the XML element itself is specified.

The taxonomy defines a general set of concepts that are applicable in the instance documents and can be referred to using the corresponding id of the concept. A concept defines each of the following aspects in separate files (XBRL Deutschland e. V., 2021b):

Definition for logical relationships of concepts that are not computational

Calculation for documentation of mathematical relationships between concepts

Reference for references to external sources of information (such as paragraphs of law)

Label for plain text names of concepts whereas multiple languages might be supported

Presentation for a standard form of presentation of financial statements

To answer the research questions, especially the translation labels of concept ids into natural language is of importance, as they can be used as a link between XBRL instance file and text reports. To determine the plain text label for the previous example concept (*taxonomy:liab*) in German language, the taxonomy extract from the German label file in Listing 2.2 is used and the XML element with the matching id holds the searched value.

Listing 2.2: XBRL label example

```
1 <label id="liab" lang="de">Passiva</label>
```

2.2.2. Standard taxonomies

Different reporting requirements by e.g. national legislation result in specific taxonomies (XBRL International Inc., 2021a). The XBRL Taxonomy Registry from XBRL International Inc. (2021b) lists a total of 51 taxonomies worldwide that are applicable in different parts of the world in order to comply to national reporting standards each. Two taxonomies tailored to German law are listed, as well as numerous for internationally recognized standards that may also be applicable in Germany, depending on the company and disclosure requirements. However, these figures only provide an overview, as this taxonomy registry is not legally binding.

The taxonomies for meeting the legal requirements in Germany are published by the Federal Ministry of Finance³. They are the concrete implementation for the form of publication of the financial report required by section 5b of the German Income Tax Act (EStG) stated as the so-called officially prescribed data format. The current version is taxonomy version 6.5 dated April 14, 2021 and it is publicly available for download (Bundesministerium für Finanzen, 2021). The latest official taxonomy package for Germany consists of the following modules (Rechenzentrum der Finanzverwaltung des Landes Nordrhein-Westfalen (Körperschaft des öffentlichen Rechts), 2021):

- GCD module for base data (de-gcd)
- Core taxonomy for a broad range of organizations (de-gaap-ci)
- Supplementary taxonomy for various industries subject to regulations (de-bra)
- Special taxonomies for banks, insurance companies and payment institutions (de-fi, de-ins, de-pi)

2.3. Federal Gazette financial statement publications

As described in section 2.1.2, annual financial statements are published in the Federal Gazette⁴. The annual financial statements for companies subject to publication require-

³<http://www.estuer.de/#finanzantrag>

⁴<https://www.bundesanzeiger.de/>

ments are available for public inspection on the website of the Federal Gazette (Bundesanzeiger Verlag GmbH, 2022b). Two different forms of presentation are usually available for each annual financial statement: A specific presentation from the Federal Gazette and the official publication files.

2.3.1. Specific presentation from the Federal Gazette

The specific presentation is manually created by the Federal Gazette and is the unofficial version, which includes in full all report components subject to disclosure in accordance with section 325 (1) HGB in one file. The file is delivered in the browser as an HTML file (see appendix A.1 for an example excerpt) and consequently contains the entire text of the annual financial statements as well as all tables, figures, etc. in semi-structured HTML data form. However, the XBRL instance files are not included in this unofficial version, so that these must be obtained from elsewhere.

2.3.2. Official publication files

The official version of the financial statement is published by the company providing the information itself and is presented by the Federal Gazette without being processed further. The version usually consists of an XHTML file viewable in the browser (see appendix A.2 for an example) and the taxonomy files applied in the financial statements (see appendix A.3 for an example). In addition to the complete report, the XHTML may also contain the XBRL file in embedded form. However, this is not the standard case and cannot be taken for granted. Alternatively, the XBRL instance file can also be submitted separately to the Federal Gazette and then this file is not published. From a legal point of view, this is not a problem, as the XBRL files do not contain any information that is not also published in the XHTML file in the form of the text report. However, with interest in structured data, this is a challenge for this work. It is not clear beforehand for which companies the XBRL instance file is embedded in the XHTML file. Furthermore, since the files are published directly by the companies, the XHTML cannot be assumed to all follow the same instantiation logic, unlike the specific version of the Federal Gazette. For example, the element tags for certain fields in the XHTML document are named differently depending on the company, which makes them difficult to process by machine. The creation of a dataset where the XBRL instance file and text report are present in each case becomes hardly solvable with the Federal Gazette website as a result.

2.3.3. Protection against data crawling

In addition to the specific problems with the different publication types, the two share another problem that prevent the creation of a dataset from the publicly viewable data of the Federal Gazette. The Federal Gazette does not offer an API, so each individual financial statement must be searched for and manually accessed via the Federal Gazette

search. This is not practical for the amount of data needed. This process could possibly be automated by a machine-controlled browser, however the Federal Gazette protects itself from this by putting the publication files behind a captcha query. Therefore it is not possible to obtain the dataset automatically.

2.3.4. Validatis data purchase

Validatis⁵ is a subsidiary of the Federal Gazette and offers the purchase of company data (Bundesanzeiger Verlag GmbH, 2022a). The service provider has access to all files submitted by the companies as part of their financial statements. Accordingly, the text report in structured XML form and also the associated XBRL instance files in a uniform format can be acquired for a large number of annual financial statements. The company PricewaterhouseCoopers GmbH WPG has acquired one dataset of this and kindly made it available for processing in the context of this master thesis. The raw data may not be published.

According to the data service provider, the purchasable data contains only financial statements that can be mapped using the HGB standard taxonomy (de-gaap-ci). The dataset might therefore not contain, for example, annual financial statements from companies in the finance or insurance sectors if they applied the special taxonomy. Large public companies that prepare their accounts exclusively according to international standards such as IFRS may also be excluded if they have not also voluntarily prepared or had to prepare HGB financial statements and the operators of the Federal Gazette have not done it manually themselves.

2.4. String matching

To annotate financial entities in the text, the respective labels provided by the XBRL taxonomy are searched in the text. The labels are strings that hold a short token sequence. Since the labels are not necessarily used exactly the same in the text, string matching is necessary. The task of string matching is to identify two different token sequences that reference the same real world object (Doan et al., 2012, p. 95) and is exactly what is needed for the first research question.

With typos (e.g. *liablities*), format differences (e.g. *01/01/2021* vs. *2021-01-01*), abbreviations (*liabilities to banks* vs. *ltb.*), synonyms (*liabilities to banks* vs. *credit institution debts*), and swaps in token order (*liabilities to banks* vs. *bank liabilities*), there are numerous challenges in string matching (Doan et al., 2012, p. 95). The solution to approach these are similarity measures between two token sequences.

⁵<https://www.validatis.de/>

2.4.1. Set-based similarity

Set-based/token-based approaches consider strings as sets of tokens. The total string is thus split into individual tokens and the similarity describes the similarity of the two comparison sets. The overlap measure as a similarity measure gives the number of common tokens of two sets. To normalize this value, a division by a constant k can be performed. This is called the Common Neighbor Score, although a suitable value for k is difficult to determine when many comparison sets are of different sizes. The Jaccard similarity measure calculates similarity with the number of common tokens divided by the number of unique tokens of both sets. The value is thus already normalized. An extension to Jaccard is Adamic, where individual tokens are assigned a weight. Tokens occurring in many sets get a lower weight than tokens occurring less often. (Doan et al., 2012, p. 104 ff.)

2.4.2. Sequence-based similarity

Sequence-based similarity measures build on a measurement of the cost of transforming a sequence A into sequence B . The Levenshtein distance is a well-known measure of edit distance. Each possible operation to get from A to B is assigned a cost. The operations per letter are insert, delete, replace and are given a cost factor of 1. If the letters at location n are the same for A and B , there is no cost (0). Thus, for identical sequences, costs of 0 are incurred, for completely different sequences, costs equal to the number of letters of the longer string are incurred. The costs can be normalized by dividing the edit cost by the length of the longer sequence. 1 minus this cost is a measure of the similarity of two sequences. Levenshtein can be extended by Affine Gap. Here, lower costs are incurred for making contiguous changes to a sequence. There is a cost for opening a gap, but the subsequent insertion in an opened gap is not as expensive as the opening. Thus, there is a lower impact if, for example, a complete word has to be inserted from A to B . (Doan et al., 2012, p. 96 ff.)

Sequence-based approaches have the major disadvantage that word swaps are not compensated for. The edit distance from A : *assets and liabilities* to B : *liabilities and assets* is therefore very high, although the terms refer to the same financial entity. Hybrid approaches combining the advantages of set-based approaches and sequence-based approaches can mitigate this problem. Hybrid approaches decompose a sequence into a multiset of tokens. A sequence-based similarity measure is then applied to the individual tokens. An example of such an approach is the Extended Jaccard. In contrast to the original Jaccard approach, not only exactly equal tokens are counted as equal, but also tokens classified as similar with a threshold by a sequence-based approach. (Doan et al., 2012, p. 106 ff.).

2.4.3. Word embedding similarity

The problem of synonyms pointed out at the beginning is the only one that cannot be solved with the presented approaches, which is why another approach has to be chosen when tackling this challenge. The chosen approach has to provide a similarity measure based on semantics, not based on characters. With word embeddings, tokens are projected into a continuous semantic space, while the position in the space is represented via a vector. Semantically similar tokens are close to each other in this space, although they may have a completely different character sequence (Tomas Mikolov et al., 2013). To obtain the similarity of two vectors the cosine similarity is calculated. If the angle between two vectors is 0 (they point in the same direction in the semantic space), the cosine is 1. For the vector representation of token sequences (n-grams), for example, the average value of the individual tokens can be calculated to represent a token sequence in a single vector. To increase the expressiveness of such vectors, additionally a weighting can be given to each token when calculating the mean value. Thus, frequently occurring terms with possibly little importance on the semantics can be assigned a lower weight to give the other tokens stronger influence on the vector of the n-gram (Kenter and de Rijke, 2015). Unimportant tokens with regard to the semantics of a token sequence, so-called stopwords, can alternatively be removed completely. There are several architectural approaches to generating word embeddings, three of which are briefly presented.

CBOW

CBOW (continuous bag-of-words) is a neural network that receives a set of surrounding tokens for the token to be predicted as input during training. So, for the prediction of the token *weekend* in the sentence *The weather on the weekend is expected to be cloudy*, the CBOW model would receive a list of tokens containing the tokens *weather, on, the, is, expected, to*. More tokens can be passed to the left and right of the target token depending on the size of the window. The model builds the sum of the vectors for the passed tokens to predict the target word. (Tomás Mikolov et al., 2013)

Skipgram

The Skipgram neural network architecture, unlike CBOW, takes only one token as input and predicts the target token being searched for based on the input vector. In the same example as before, the token *weekend* would be predicted by a token nearby, e.g. *weather* or *the*. (Tomás Mikolov et al., 2013)

GloVe

The GloVe (global vectors) approach constructs a global word-word co-occurrence matrix for tokens in the training corpus. The global word co-occurrence statistics is the main difference to CBOW and Skipgram, where only the local context is taken into account. This

means that when applying CBOW or Skipgram in the given example it is not possible to derive whether the word *the* is meaningful and special in the context of the word *weekend* or if it is just some kind of stopword or otherwise semantically unrelated to it. The global approach of GloVe addresses this issue. (Pennington et al., 2014)

Pre-trained word vectors are available for both the CBOW/Skipgram and the GloVe approach. FastText offers pre-trained vectors trained with the CBOW approach for 157 different languages⁶ (Grave et al., 2018), including German. There is also a script available for training on a specific corpus or using Skipgram instead of CBOW. The authors of GloVe also offer ready-to-use pre-trained vectors⁷ (Pennington et al., 2014).

2.5. Linguistic parsing

A natural language parser extracts the structural constituents from a sentence like grouped phrases, which word in the sentence is the verb, which is the corresponding subject and object etc. (Stanford NLP Group, 2022).

2.5.1. Part-of-speech parser

Part-of-speech (POS) tagging is a task of linguistic parsers which aims to assign a descriptor to each input token of a sentence such as noun, verb, participle, article, pronoun, preposition, adverb and conjunction (Voutilainen, 2003, p. 220). spaCy⁸ offers a POS tagger as part of their natural language processing framework which will be used later in the experiments.

2.5.2. Dependency parser

Dependency parsers establish head-modifier links between tokens in a sentence each of which gets labelled with a grammatical function (Carroll, 2003, p. 235). Figure 2.2 shows an example to visualize this using an example sentence processed by a dependency parser⁹.

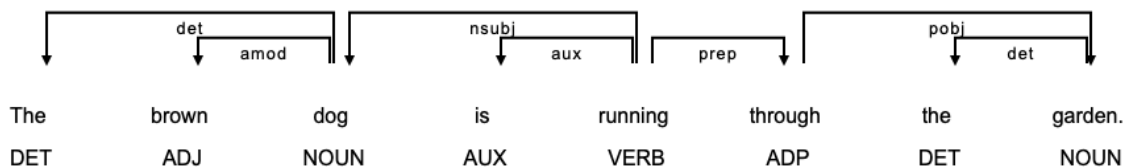


Figure 2.2.: Example sentence processed by a dependency parser

⁶<https://fasttext.cc/docs/en/crawl-vectors.html>

⁷<https://nlp.stanford.edu/projects/glove/>

⁸<https://spacy.io/>

⁹<https://explosion.ai/demos/displacy>

Each token in the sentence gets assigned at least one labelled relationship to another one. The token *brown*, for example, is a modifier for the token *dog*. *Dog* is also the subject of the sentence which can be derived by its relation to the verb of the sentence. spaCy offers not only POS tagging but also dependency parsing and is used for this purpose as well.

2.6. Entity detection

Named entity detection/recognition (NER) is an information retrieval (IR) task of identifying expressions in unstructured text that refer to entities like peoples, places, organizations, companies, etc. NER is divided into two tasks of which the first is to identify named entities in text and the second is to classify them correctly (Mansouri et al., 2008). In the context of this work NER is performed to detect financial entities and financial values. In this case a financial entity is an abstract concept which has a semantic, financial meaning, a computation rule to calculate the value, relationships to other entities and an obligation to be reported in a financial statement. For example, the financial entity *assets* is the sum of all asset items and at the same time the balance sheet total of a company. The calculation is defined by law and reflected in the XBRL taxonomies.

In the sentence *At the end of the financial year, liabilities to banks amounted to TEUR 6,000*, the expression *liabilities to banks* is a financial named entity that is supposed to be identified in the experiments carried out in this work. The value *6000* is the associated financial value that should also be recognized. With financial entities and financial values, there are thus two types of entities.

2.7. Relation extraction

Given a set of named entities the task of relation extraction is to identify the type of relationship between them (Bach and Badaskar, 2007). Relation extraction is a core functionality to create knowledge graphs and is also used in structured search, question answering, summarization, etc. (Huang and Wang, 2017). Many relation extraction approaches focus on binary relationships like *located_in*(*Hamburg*, *Germany*) where both *Hamburg* and *Germany* are entities and the relationship is a directed instance of *located_in*. In the context of this work relation extraction is used to find matching pairs of financial entities and financial values. In the sentence *At the end of the fiscal year, liabilities to banks amount to EUR 6000, of which EUR 3000 are long-term liabilities with a maturity of more than one year*, there are now two entities (*liabilities to banks* and the sub-entity *long-term liabilities with a maturity of more than one year*) and two financial values (6000 and 3000). The amount of possible entity-value pairs is now no longer only one which is why the correct relationship between entities and values has to be captured in the training data.

2.8. Distant supervision

Distant supervision is an approach to relation extraction that does not require annotated sentences. For distant supervision, structured semantic databases are used, which store entities and their relations to each other. The entities from the knowledge base are searched in natural language sentences and all hits can be used as training examples for a classifier. The disadvantage of this method is the expected amount of noise in the extracted training examples. The occurrence of two entities within a sentence is not a definite indication that their relationship is actually expressed in this sentence. Consequently, the quality of the training data suffers. On the other hand, it is more efficient than manual annotation of sentences. (Mintz et al., 2009)

Distant supervision is used in this work to find financial information, which is available in a structured key-value XBRL format, in natural language sentences and to use these sentences as training samples for a classifier that detects financial entities and their associated values in text without the structured XBRL format available.

2.9. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a language representation model introduced by Devlin et al. (2019) leveraging the transformer neural network architecture which was introduced by Vaswani et al. (2017) and was originally designed for machine translation.

Before BERT was introduced, machine translation was traditionally handled with Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, which are a specific type of RNNs (Sherstinsky, 2020). RNNs solve this task by processing an input sequence of tokens one token at a time and sequentially produce the translated tokens. The order of words is indispensably important in a sentence for the semantic meaning and RNNs keep track of the order by sequential processing. This is also one of their biggest disadvantages since sequential processing does not allow for parallelization during training. On top of that RNNs do not capture the bidirectional context of words and have problems handling long sentences because the corresponding context gets lost over time. LSTMs also do not capture the deep bidirectional context because they only process the sentence context once from the beginning and once from the end separately and then concatenate this representation. (Hochreiter and Schmidhuber, 1997), (Cho et al., 2014)

The transformer neural network architecture (Vaswani et al., 2017) consists of an encoder and a decoder where the encoder takes the input sequence (all tokens simultaneously), generates embeddings from it and the decoder takes this representation to gener-

ate the next output word until the next predicted word is the end token. Since BERT is not explicitly intended to solve the machine translation task but to generate context-sensitive token representations the decoder is not necessary. Only the encoder is needed because the output of this is the desired language representation. The token embeddings then can be used on a variety of tasks instead of explicitly being applied to machine translation. (Rush, 2018)

The goal of the training process for BERT (Devlin et al., 2019) is to understand the semantic meaning of language and context. Therefore BERT utilizes masked language modeling and next sentence prediction. The task in masked language modelling for the BERT model is to predict masked tokens in a sentence based on the context and the task in next sentence prediction is, based on two input sentences, to predict whether the second input sentence actually follows the first one.

During training of BERT as a general language model, pairs of sentences are passed to the model with some of the tokens randomly being masked. The input tokens are encoded using 1) token embeddings, 2) segment encodings and 3) position encodings. In the original paper, WordPiece 30k (Ravichandiran, 2021, p. 65) is used for the token embeddings. The segment encoding keeps track of which token belongs to which sentence (first or second) and the position encoding stores the numerical position in the sentence for each token. This way the information about word order is kept in the data itself rather than in the structure of the model network which is a great improvement over RNNs. These three input information vectors are added up and passed to the model. (Rogers et al., 2020), (Rush, 2018)

The output for each sentence pair is a binary value whether the sentences belong to each other (second actually follows first) and a token embedding for each token as well as one overall sentence representation. Each token vector is connected to a softmax layer which has the size of the vocabulary (30k in case of WordPiece 30k) which is used to predict one single token from the vocabulary. The loss function that is minimized during training compares the models output with the actual expected token and backpropagates the error with a cross-entropy loss function for the masked tokens of the input sentence. Inside the model the core part consists of a self-attention neural network structure. Self attention is a special instantiation of the attention mechanism which is crucial for transformer networks. Attention allows the model to look at every single token in the input sequence when making a decision on the output translation sequence. This is how the context awareness is created in transformer networks. Self-attention turns the attention on the input sentence itself rather than on the output sequence. Since BERT is not designed for translation but for obtaining a language model there is no attention necessary from an input sequence to an output sequence and therefore the attention is turned on

the sentence itself. For each token the model learns which surrounding tokens are crucial in order to understand the word in its context. (Rogers et al., 2020), (Rush, 2018)

The contextualized word embeddings obtained from BERT will serve in this work as the core part of the model to identify entities and their relations in natural text. Pre-trained BERT embeddings from large corpora are fine-tuned with an additional layer to network architecture in order to learn how embeddings for financial entities and financial values look like.

3. Related work

The related work for this thesis is divided into two parts. On the one hand, related work on the more general issue of entity detection and relation extraction is considered, and on the other hand, prior work on the very concrete issue of extracting financial entities and their values from financial statements in the context of automated auditing is examined.

3.1. Entity detection and relation extraction

Entity detection and relation extraction can be considered as two separate tasks in a consecutive pipeline, however state-of-the-art (SOTA) results have recently been achieved with approaches that combine both tasks in a joint fashion (Nasar et al., 2021). Joint entity detection and relation extraction statistical models, in contrast to two separate models, are able to detect underlying dependencies between entities and relations, and a combined end-to-end error function can be used so that errors in the entity detection step are not simply propagated to the relation extraction step (Gupta et al., 2016). The related work analysis therefore will focus on joint approaches.

Relevant for this thesis are approaches tackling the challenge of intra-sentence joint entity detection and binary relation extraction with supervised learning. Entity detection and relation extraction on document level without the local restriction for entities to occur in the same sentence is a different problem (Eberts and Ulges, 2021) and is not considered in this work.

Taillé et al. (2020) provide an up-to-date overview of approaches which covers papers on joint entity detection and relation extraction up to 2020. According to this survey, publications from the recent past tend to use pre-trained language models such as BERT and ELMo in order to obtain contextualized word and span representations which are then used to extract entities and relations. Static word embeddings and pooled character embeddings hardly play a role in current releases. The same applies to handcrafted features as well as part-of-speech tags and dependency tags, which originate either from external taggers/parsers or from manual annotations. However, they were a common part of the approaches of the time a few years ago.

Zhong and Chen (2021) present a very typical approach by using two independent encoders, where the entity detection model has as main task to construct the input for the

relation extraction encoder. They also emphasize in their work that information about the entities should be included in the relation extraction model as early as possible and with global context information. Wang and Lu (2020) also use two distinct encoders in the learning process, but they formulate the problem as a table-filling problem and use a table-encoder and a sequence-encoder. The two encoders can help each other because they each have additional information from the other task. Formulating end-to-end entity detection and relation extraction as a table-filling problem is generally a common approach. The tokens of the sentence are plotted on both the x and y axis of the table. On the diagonal the tokens meet each other and their entity type is classified. In all other cells the relation between the two crossing tokens is classified, so that entities and relations can be mapped using a single table. Ma et al. (2022) implement this approach by using contextualized word embeddings, eliminating the need for complicated manual features. The relations in the table are all predicted then in one step, without the use of search strategies or prior predictions. Earlier approaches use an alternative to contextualized word embeddings and therefore resort to other approaches to populate the table. Gupta et al. (2016) propose table filling multi-task recurrent neural networks for this purpose. Wang et al. (2021) argue that having two label prediction spaces (list of predictable entity types and list of predictable relation types) like in the mentioned approaches prevents the model from learning the interdependencies between entities and relations. Instead they apply a unified label space to the table-filling problem. Contrary to the table-filling approach, Li et al. (2019) formulate the problem as a multi-turn question-answering problem. The extraction of entities and relations is thus formulated as a task to identify answer spans from the given context. The question is used to encode information about the entity class or relation class being searched for, and pre-trained machine reading comprehension models can be applied. The disadvantage of this approach is that pre-fabricated patterns have to be formulated for the questions, the generality of which is naturally limited.

Many of the current state-of-the-art approaches share a similar, span-based architecture meaning instead of single tokens, token sequences (spans) are classified as entities and the entities are related to each other afterwards. Span-based models tend to perform stronger in recent experiments because the sequential decoding of token-level features produces cascading errors and they fail to implement span-level features as well as overlapping entities since every token can get assigned only one single tag (Dixit and Al-Onaizan, 2019). There are different ways to create a single fused representation for a token span and also different approaches to combine the representations of two token spans to classify their relationship to each other, especially which information about the entities to feed to the relation classifier and how to encode them. Zhong and Chen (2021) first obtain contextualized word embeddings (e.g. BERT) for each token and subsequently span representations are calculated as the average of the first and last token for

each potential entity span. Using a feedforward network the entity class is predicted from this representation. The attached relation model takes a pair of spans as input and predicts the relation type based on that. Instead of simply reusing the span representations in every pair of spans each span gets a marker embedded to the encoding whether it is the object or the subject of that relation and also which type of entity has been predicted. Therefore a single span can get different representations based on which pairing span it is connected to. Sharing the same contextual representations between different relation spans is not helpful for the model the authors argue. Baldini Soares et al. (2019) also insert subject and object markers as well as entity boundary tags and feed them to the relation extraction model. The weakness of relation classification approaches that accept only a single pair of spans independently at a time is that they cannot represent the interrelation between multiple spans. To address this problem Ye et al. (2022) introduce a span-representation approach which uses a neighborhood-oriented marker packing strategy to integrate the neighbor spans to model the entity boundary information. For each subject not every single possible object is processed independently by the relation extraction step but all objects at once in order to model the interrelation between same-subject entities. Luan et al. (2018) present a span-based multi-task setup with a unified model where parameters of multiple low-level tasks are shared among each other. By jointly modeling all the possible spans and their relations the propagation error between the two tasks is minimized. The successor of this approach is presented by Luan et al. (2019) and uses a graph structure where entities are represented as nodes and the edges capture confidence-weighted relation types. This approach in turn was later modified by Wadden et al. (2019) to replace the BiLSTM encoder with BERT making it a transformer-based and span-based approach. Eberts and Ulges (2019) also implement a span-based attention model called SpERT which renounces markers for context representation and focuses on a light-weight reasoning on BERT embeddings. The work demonstrates the strength of contextualized word embeddings and negative samples in the training process. Santosh et al. (2021) build on this work and investigate the role of encoded part-of-speech tags for the entity detection step and encoded prediction logits from the entity detection step for improving the relation extraction step. The authors focus on the scientific domain using the SciERC (Luan et al., 2018) and ADE (Gurulingappa et al., 2012) datasets. SciERC consists of six scientific entity types (e.g. method, task, metric) and seven relation types (e.g. compare, used-for, evaluate-for) while ADE consists of two entity types (adverse-effect and drug) and one single relation (adverse-effect). Their work performs slightly better than the plain SpERT model on these two datasets.

3.2. Financial statement information extraction

Kamaruddin et al. (2009) deal with a very similar issue as in this work. They extract key performance indicators (KPI) from financial statements, which is nearly equivalent to fi-

nancial entities and their associated values. They use a purely rule-based approach and focus only on three entities that are most relevant from their point of view. Accordingly, to a lesser extent, the question in this work is exactly the same, but the approaches do not correspond to today's SOTA, which is due to the state of research at that time. Brito et al. (2019) also introduce a financial key value extraction tool which is meant to extract key performance indicators from financial reports and therefore is very similar to the goal of this work. The system was trained with human annotated data and additionally uses a rule-based extraction approach. In the first step, a web crawler is used to download financial reports from companies' websites after publication. Tables and potentially relevant text passages are then identified in the document. The documents are processed as images to detect tables within them using recurrent neural networks. The tables are searched for synonyms and exact matches of tokens that are in the ground truth listing of financial entities which was manually created. The natural language text is split into sentences using spaCy, tokenized and the dependency tree is created. Then, all sentences that do not contain a numeric value are discarded. For each numeric value in the remaining sentences, special tokens are extracted from the dependency tree (e.g. root, parent etc.). The tokens are then represented numerically using word2vec representations of the numeric token and the selected ones from the dependency tree. The word2vec model was specifically trained on financial reports. A tree-based classifier then is learned to predict a label from the financial entity list for a given word2vec input representation. The paper seems to be aimed at practitioners by its structure and topic focus, and unfortunately no evaluation is presented. The performance of the tool is therefore not publicly known. However, since the approaches for extracting entities with respect to section 3.1 tend to no longer match those of SOTA approaches, the performance is presumably behind the performance of current joint entity detection and relation extraction work. For example, word2vec is used instead of contextualized word embeddings like BERT, and the use of the root element of the dependency tree becomes difficult to use for multiple financial entities with financial values in a single sentence.

Sifa et al. (2019) present a software tool suite for automated auditing with machine learning approaches. The focus of their work is a legal matcher which matches text passages in financial statements to legal requirements by the legislator. This helps auditors to automatically ensure the completeness of a financial statement. They compare different approaches like n-grams, bag-of-words and neural language models for this. The software suite also offers a component to make sure of the internal consistency of the financial statement (e.g. no contradictions between financial values that are presented in both the text and the tables). Unfortunately the authors do not present the performance of this component and also do not provide any details on the approach used for this feature. To ensure the document-internal consistency of the financial statement, the financial entities and their corresponding values have to be extracted from text which is very sim-

ilar to this work.

Chapman et al. (2021) introduce an approach to generate financial statements, in particular the natural language text part, from table data. While earlier approaches in this area focus on pre-formulated sentence patterns which are filled with the table data, the authors use a transformer network to overcome the problem of missing creativity. While they have to deal with some similar domain-specific challenges like financial numeric values, currency tokens, etc., the goal of generating text instead of extracting structured information from it is the counterpart of this work.

3.3. Research gap

On the one hand, specific approaches to extract financial information from financial statements are limited, do not use SOTA methods, and lack open source code and transparent evaluation. Entity detection and relation extraction as an abstraction of the problem is on the other hand primarily not specifically tailored to domain-specific challenges of financial reports and also not measured against datasets from this domain. Furthermore, the approaches are based on manually created training data.

This work serves to attempt to apply entity detection and relation extraction to information extraction from financial reports. For this, SOTA methods shall serve, be adapted to the challenges of the specific domain and the training dataset shall be generated with distant supervision without manual work.

4. Joint entity detection and relation extraction approach

This chapter presents a model for entity detection and relation extraction and therefore provides the foundation for the second research question. The model is extended and evaluated on a public dataset in order to measure the effects of the changes. The code of the customized model is available on GitHub¹.

4.1. Span-based entity and relation transformer

The basis for the detection of entities and relations is the span-based entity and relation transformer (SpERT) model (Eberts and Ulges, 2019). It achieves state-of-the-art results and provides an open-source code repository² which allows for customization. Figure 4.1 illustrates the architecture of the original model and this overall section refers to the original paper.

The setup first takes a sequence of tokens as input to the BERT tokenizer resulting in a sequence of byte-pair encoded (BPE) tokens $(e_1, e_2, \dots, e_n, c)$ where c refers to the $[CLS]$ token. Byte-pair encoding ensures that tokens are split into sub-tokens in case of infrequent and out-of-vocabulary words. This way the vocabulary is limited to a given size and after tokenizing there are no out-of-vocabulary words (Sennrich et al., 2016). The representation of sub-tokens is demonstrated in figure 4.1 below the BERT model where the token *TEUR* was split into the available sub-tokens *T* and *EUR*. After passing the BPE tokens through the BERT model, each token gets a contextualized embedding and the overall sentence embedding is represented by the embedding of the $[CLS]$ token at the beginning. The $[SEP]$ token indicates the end of a sentence.

After obtaining the contextualized word embeddings for each BPE token, the SpERT model constructs every possible token span $s := (e_i, e_{i+1}, \dots, e_{i+k}) \in S$ up to a configured length k less than or equal to the amount of BPE tokens. The BPE token embeddings of each span representation then are max-pooled to a fused token span representation $f(e_i, e_{i+1}, \dots, e_{i+k})$. Max-pooling all the tokens of the corresponding span outperformed summing and averaging the representation values in the authors experiments. This rep-

¹<https://github.com/farausch/spert>

²<https://github.com/lavis-nlp/spert>

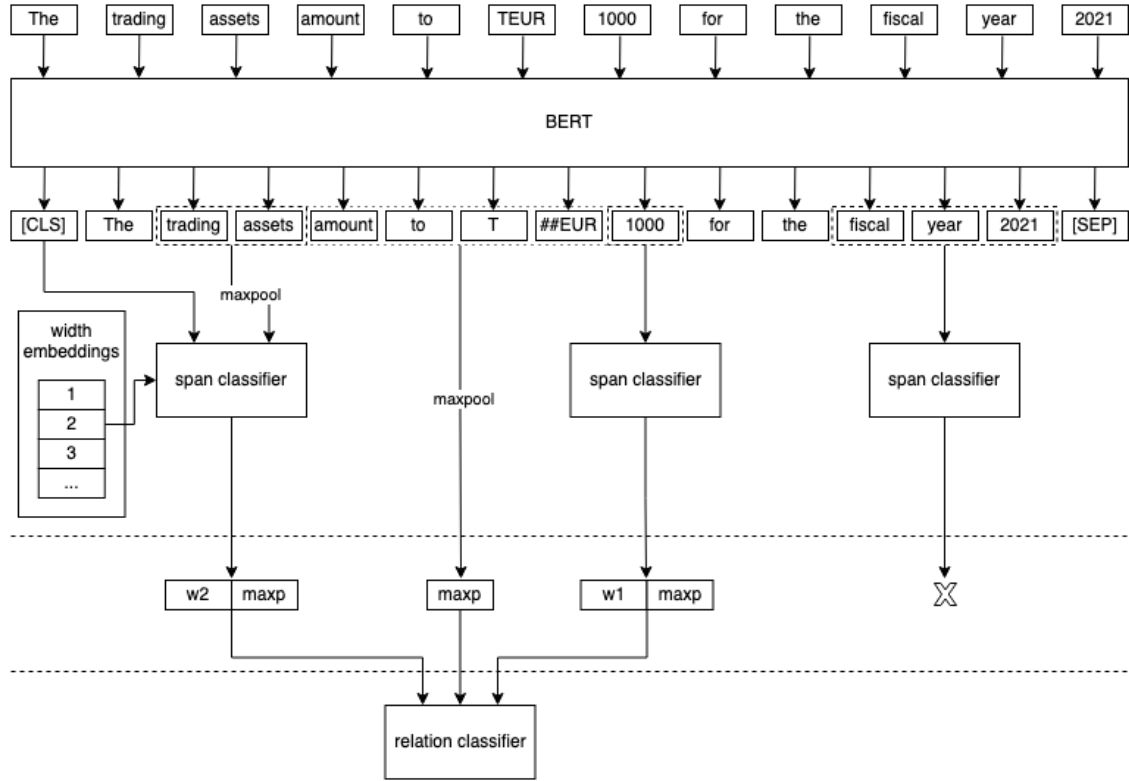


Figure 4.1.: SpERT architecture according to Eberts and Ulges (2019), slightly modified to a financial example

representation is fed to the span classifier concatenated together with the overall sentence embedding represented by the $[CLS]$ token and a width-embedding $w_n \in W$. The width-embedding encodes the length of the token span by looking up a fixed-size encoding from a matrix which contains encodings for each possible span length. These embeddings are learned during the model training process and play a vital role in the span classification step. The length of a token span is a crucial information for classifying entities, especially for ruling out spans that are too long, and was first introduced for entity coreference resolution (Lee et al., 2017). It has shown to contribute significantly to the resolution accuracy (Kahardipraja et al., 2020). Wadden et al. (2019) also use width embeddings and feed it to their model together with a concatenation of the left and right endpoint representation of a span which is slightly different than using the max-pooled representation of the whole span. In the example in figure 4.1 the span to be classified (*trading assets*) has a length of two which is why the width embedding w_2 is looked up and fed to the classifier. Concatenating these three terms leads to the following span representation whereas \circ denotes concatenation and f represents the max-pooling fusion function:

$$e(s) := f(e_i, e_{i+1}, \dots, e_{i+k}) \circ w_n$$

Finally, this representation is extended by the overall sentence representation kept in the

[CLS] token which ends up in the following input passed to the span classifier:

$$x^s := e(s) \circ c$$

This representation is fed into the softmax classifier which returns a posterior for each entity class including the *none* class:

$$y^s = \text{softmax}(W^s \cdot x^s + b^s)$$

The span classification step is followed by the span filtering step. Each span which is not classified as an entity is discarded (like *fiscal year 2021* in the given example) while all spans that are classified as entities (in this case *trading assets* and *1000*) are then combined as candidate pairs. Classified spans longer than a threshold (pre-configured value is 10) are also discarded in order to limit the complexity.

The relation classifier for each candidate pair takes as input a concatenation of both of the candidate entity spans with embeddings (here w_2 and w_1), both max-pooled span representations and the context: $x^r := e(s_1) \circ c(s_1, s_2) \circ e(s_2)$. The context c is the max-pooled fusion of all of the token representations that are between the two candidate spans (*amount to TEUR* in this example). Given this input the relation classifier will predict the relation type between the two entities using a single-layer classifier:

$$y^r := \sigma(W^r \cdot x^r + b^r)$$

whereas a high value in the sigmoid layer σ of the size of all relation types indicates that the corresponding relationship is present. Since relations can be generally asymmetric, the relation classifier must be called twice for a candidate pair so that each entity is once on the left and once on the right of the context (s_1 becomes s_2 and vice versa).

During the supervised training process both the width embeddings and the span/relation classifiers are learnt. The joint loss function is defined as the sum of the cross-entropy loss over the entity classes and the binary cross-entropy loss over the relation classifier and is the key element in joint entity detection and relation extraction approaches. Besides the positive entity span and relation examples from the ground truth training data, the training algorithm creates negative examples to learn from in order to increase the outcoming model performance. For the entity detection step, wrong spans are randomly created from the tokens and the model is expected to predict the *none* class. Regarding relation extraction, the authors create so-called strong negative examples meaning they construct only negative samples from entities that were found by the model in the first place but are not actually related.

4.2. Model extension

The authors of SpERT refrain from using linguistic information about individual tokens within the model and rely on BERT embeddings only to classify entities and their relationship to each other. In this section linguistic features are introduced to the model.

4.2.1. Part-of-speech tags

The model at the end of this thesis is intended to reliably recognize financial entities and financial values, and since financial values are always numeric in nature, explicit part-of-speech tags can be helpful to the model especially for this case. The model can thus learn that a token without a numeric part-of-speech tag cannot be a financial entity in any case. Table 4.1 shows the part-of-speech tags for the introduced example sentence.

Table 4.1.: Part-of-speech tags for example sentence

Token	POS tag	Explanation
The	DET	Determiner
trading	NOUN	Noun
assets	NOUN	Noun
amount	VERB	Verb
to	ADP	Adposition
TEUR	NOUN	Noun
1000	NUM	Numeral
for	ADP	Adposition
the	DET	Determiner
fiscal	ADJ	Adjective
year	NOUN	Noun
2021	NUM	Numeral

The example reveals some generally valid hypotheses. In addition to the numerical nature of financial values, for example, financial entities also always require a token sequence that contains at least one noun. In this case, there are two nouns attached to each other, which is already a strong indicator for an entity. The tags can also be helpful in extracting the relationships. In this example, the only verb in the sentence is in the actual context area, which is used to classify the relationship.

The part-of-speech tags are integrated into the model by passing each sentence through spaCy’s part-of-speech tagger when the dataset is read. Each token is thus assigned a tag. This additional input feature is encoded with one-hot-encoding before the tokens are passed to the span classifier or the relation classifier. In other words, the BERT embedding of each token is extended by a vector of the length of the number of all possible part-of-speech tags where all values are set to 0 except the position of the corresponding part-of-speech tag which is set to 1. The BPE token sequence representation now

becomes $(e_1 \circ p_1, e_2 \circ p_2, \dots, e_n \circ p_n, c)$ instead of $(e_1, e_2, \dots, e_n, c)$ where \circ again denotes concatenation and p is the one-hot encoded part-of-speech tag. The list of all tags depends on the applied model in the parsing process. For the German spaCy transformer model which will be used later the list can be found on the model description page³. The implementation of this extension can be found on the main branch in the GitHub repository⁴.

4.2.2. Dependency tags

In addition to part-of-speech tags, dependency tags can also be useful for classifying entities and relations. Dependency trees have played an important role in relation extraction (Fundel et al., 2007), (Culotta and Sorensen, 2004) before language models replaced them. The spaCy framework provides a dependency parser and figure 4.2 shows the output of it for the financial example sentence.

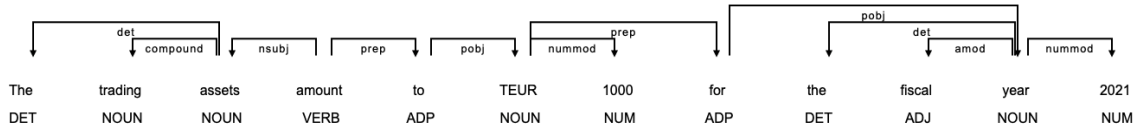


Figure 4.2.: Dependency tags for an example sentence

The figure shows that, for example, the token *trading* has the dependency tag *compound* and the token *assets* points to it. *Trading* is therefore a child of *assets* with the type *compound*. This is a very useful information in order to find the correct boundaries of an entity. The token *The* is also a child of *assets* and the model can learn that the dependency tag for determiner is not part of an entity typically and very likely not the boundary of an entity. The token *amount* is the only token in this sentence which has children but is not the child of any other token which makes it the root token of this sentence. The root token is often a strong indicator of the relationship that is existing in the given sentence like it is in this sentence as well. The financial value *1000* can also be delimited with the help of the numerical modifier dependency tag of the previous token.

The dependency tags are one-hot encoded like the part-of-speech tags which leads to a second vector of the length of all possible dependency tags appended to the BERT embeddings of every single token. Each token now has a vector representation of the size of the hidden BERT layer plus the amount of all part-of-speech tags plus the amount of all dependency tags leading to the following sequence representation:

$$(e_1 \circ p_1 \circ d_1, e_2 \circ p_2 \circ d_2, \dots, e_n \circ p_n \circ d_n, c)$$

³https://spacy.io/models/de#de_dep_news_trf-labels

⁴<https://github.com/farausch/spert>

where d is the one-hot encoded dependency tag. The list of dependency tags depends on the spaCy model and can be found again on the model description page⁵. The implementation of this extension can be found on the main branch in the GitHub repository⁶.

4.2.3. Shortest dependency path

Bunescu and Mooney (2005) hypothesize in their work that the relation between two entities in one sentence is typically exclusively captured by the tokens on the shortest path between them in the undirected dependency graph. To visualize this with the example from before the sentence is restructured in order to increase the context tokens between the two entities. Figure 4.3 illustrates the new sentence with the same semantic content.

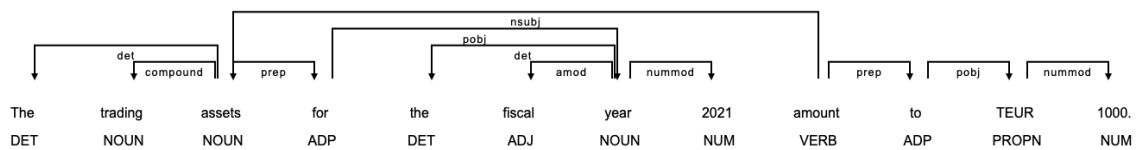


Figure 4.3.: Dependency tags for an example sentence

To make the parent-child relationships between the tokens more vivid the dependency tree is transformed into a graph visualization. The information about relations between tokens remains the same. Figure 4.4 illustrates this graph.

From the graph representation, the shortest dependency path, i.e. the shortest path between two tokens, can be read quite easily. Between the last token of the first entity *trading assets* and the first, and single, token of the second entity *1000* there are only the tokens *amount to TEUR*. All other tokens are linguistically placed between the two entities, but they are not on the shortest dependency path. According to the approach of Bunescu and Mooney (2005), all tokens that are not part of the shortest dependency path are ignored when classifying the relationship between the two entities. At least for the example shown, the approach seems reasonable.

In the implementation, this approach is realized by first deriving the dependency tree for each record with spaCy when reading in the data⁷. Later in the training process, all tokens that are not part of the shortest dependency path are ignored when forming the max-pooled span representation⁸. The model therefore is forced to restrict itself to the embeddings of the shortest path tokens during the learning process in order to eliminate noise, and better performance is expected as a result. The implementation of this exten-

⁵https://spacy.io/models/de#de_dep_news_trf-labels

⁶<https://github.com/farausch/spert>

⁷https://github.com/farausch/spert/blob/feature/dep-tree-shortest-path/spert/input_reader.py#L264

⁸<https://github.com/farausch/spert/blob/feature/dep-tree-shortest-path/spert/sampling.py#L210>

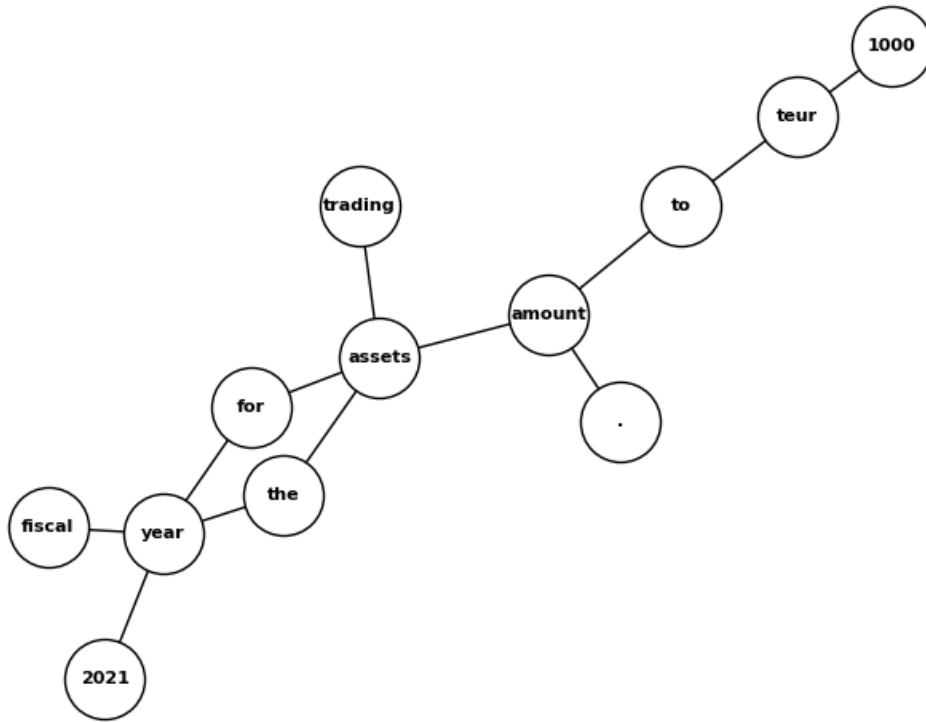


Figure 4.4.: Dependency graph for an example sentence

sion can be found on the branch *dep-tree-shortest-path* on GitHub⁹ together with the other extensions.

4.3. Evaluation

The original SpERT model as well as two modified versions are evaluated using the CoNLL04¹⁰ dataset (Roth and Yih, 2004) using the train and dev split which is the same setup like Eberts and Ulges (2019) used in their experiments. The hyperparameters¹¹ are also kept the same in order to make the results comparable. For the part-of-speech and dependency tagging spaCy was used with the English transformer model¹² which can be configured in the training configuration file. An entity prediction is considered correct if both the span and the entity type are correct. Relation predictions are correct if the relation type is correct and the predicted spans are correct by the given definition. Precision (P), recall (R) and F1 score (F) are measured. Table 6.1 shows the evaluation results.

The first row shows the micro average results while the second row states the macro average for each of the three experiments. The first experiment is a simple reproduc-

⁹<https://github.com/farausch/spert/tree/feature/dep-tree-shortest-path>

¹⁰<http://lavis.cs.hs-rm.de/storage/spert/public/datasets/conll04/>

¹¹https://github.com/lavis-nlp/spert/blob/master/configs/example_train.conf

¹²https://spacy.io/models/en#en_core_web_trf-labels

Table 4.2.: SpERT unmodified and extended version evaluation on CoNLL04 train and dev split

Model	Entity detection			Relation extraction		
	P	R	F	P	R	F
SpERT (unmodified)	88.50	90.55	89.51	73.42	68.72	70.99
	85.91	88.05	86.94	74.61	70.32	72.29
SpERT (one-hot POS, DEP and SDP)	90.20	88.69	89.44	13.15	82.94	22.71
	88.05	85.83	86.81	13.67	83.75	23.32
SpERT (one-hot POS, DEP)	89.66	89.99	89.82	72.39	73.93	73.15
	87.74	87.09	87.32	73.39	75.08	73.89

tion of the experiment originally conducted by the authors with no adjustments. The reproduced results are within the expected range of statistical difference due to random initialization of the model weights. Passing part-of-speech and dependency information to the model (second and third experiment), the entity classification F1 score did not change noteworthy. For the relation extraction step, passing the shortest dependency path tokens only instead of the whole local context leads to a better recall but a huge decrease in precision which is why this modification is not used later for the financial data extraction and also why it is not merged to the main branch. Using part-of-speech tags, dependency tags and the full context, however, also helps the model to significantly improve the relation extraction recall while the precision drops to an acceptable level. The F1 scores in comparison to the original model increase by 2.16 (micro average) and 1.6 (macro average) respectively. Having access to explicit linguistic information proves to be helpful for this specific dataset.

The modified model achieves SOTA results on the CoNLL04 dataset and therefore will be used later for the extraction of financial entities and relations.

5. Distant supervision for financial report annotation

This chapter is dedicated to the automated financial entity and value annotation of sentences in German financial statements using XBRL files in the form of distant supervision. The results of this chapter provide the answer to the first research question. The source code for the method developed in this chapter can be found on GitHub¹.

5.1. Dataset

The dataset used to develop a distant supervision method consists of 5604 financial statements provided by Validatis, for each of which the structured XBRL form and the natural text are available. This means that for each text document in the dataset, the corresponding XBRL instance document exists and vice versa. The text document contains natural language, while the XBRL instance document contains structured information about companies and their financial data. Both forms of representation are embedded in XML format and will be analyzed in the following in order to be able to use them purposefully in upcoming tasks.

5.1.1. XBRL instance documents

The XBRL instance documents instantiate a XBRL taxonomy and contain meta information (e.g. a reference to the applied taxonomy, the company name, industry key etc.) as well as the set of financial facts of the organization. Listing 5.1 displays an exemplary financial fact data section excerpt from an XBRL instance document taken from the dataset.

Listing 5.1: XBRL file data section extract

```

1 <de-gaap-ci:bs.ass contextRef="id210512038457_CY_INSTANT" decimals="2"
   unitRef="id210512038457_UNIT">1066624466.94</de-gaap-ci:bs.ass>
2 <de-gaap-ci:bs.ass.currAss contextRef="id210512038457_CY_INSTANT"
   decimals="2" unitRef="id210512038457_UNIT">373668620.69</de-gaap-
   ci:bs.ass.currAss>
3 <de-gaap-ci:bs.ass.currAss.cashEquiv contextRef="
   id210512038457_CY_INSTANT" decimals="2" unitRef="
   id210512038457_UNIT">20987.70</de-gaap-ci:bs.ass.currAss.cashEquiv>

```

¹<https://github.com/farausch/xbrl-distant-supervision>

```

4 <de-gaap-ci:bs.ass.cashEquiv.bank contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">20987.70</de-gaap-ci:bs.ass.cashEquiv.
    bank>
5 <de-gaap-ci:bs.eqLiab contextRef="id210512038457_CY_INSTANT" decimals="
    2" unitRef="id210512038457_UNIT">1066624466.94</de-gaap-ci:bs.
    eqLiab>
6 <de-gaap-ci:bs.eqLiab.accruals contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">45972596.70</de-gaap-
    ci:bs.eqLiab.accruals>
7 <de-gaap-ci:bs.eqLiab.equity contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">711940703.16</de-gaap-
    ci:bs.eqLiab.equity>
8 <de-gaap-ci:bs.eqLiab.equity.capRes contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">615516813.65</de-gaap-ci:bs.eqLiab.equity.
    capRes>
9 <de-gaap-ci:is.netIncome.tax contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">6059362.00</de-gaap-
    ci:is.netIncome.tax>

```

Each financial fact (lines 1 to 9) deals with a single financial value for the organization. Each fact references a concept via the tag id (e.g. *de-gaap-ci:bs.ass* in line 1), which can be resolved to a financial entity with a computation logic, plain text label, etc. using the associated taxonomy. Following the tag id, the *contextRef*, *decimals* and *unitRef* fields are reported for each fact as XML attributes. The field *contextRef* refers to a concept for the temporal context of the financial value, *decimals* indicates the number of decimal places and *unitRef* refers to a concept for the unique resolution of the unit of the financial value. Finally, the value of the XML element states the numeric value for the referenced concept (e.g. *1066624466.94* in line 1). (XBRL Deutschland e. V., 2021b)

While the tag id for the financial entity concept is resolved using a separated taxonomy file, *contextRef* and *unitRef* are decoded in the XBRL instance document itself in a dedicated section. Listing 5.2 illustrates the context section for the previous example.

Listing 5.2: XBRL file context section

```

1 <context id="id210512038457_CY_INSTANT">
2 <entity><identifier scheme="http://www.bundesanzeiger.de/ebanz">
    210512038457</identifier>
3 </entity><period>
4 <instant>2020-12-31</instant>
5 </period></context>
6 <context id="id210512038457_CY_DURATION">

```

```

7 <entity><identifier scheme="http://www.bundesanzeiger.de/ebanz">
  210512038457</identifier>
8 </entity><period>
9 <startDate>2020-01-01</startDate>
10 <endDate>2020-12-31</endDate>
11 </period></context>
12 <context id="id210512038457_PY_INSTANT">
13 <entity><identifier scheme="http://www.bundesanzeiger.de/ebanz">
  210512038457</identifier>
14 </entity><period>
15 <instant>2019-12-31</instant>
16 </period></context>
17 <context id="id210512038457_PY_DURATION">
18 <entity><identifier scheme="http://www.bundesanzeiger.de/ebanz">
  210512038457</identifier>
19 </entity><period>
20 <startDate>2019-01-01</startDate>
21 <endDate>2019-12-31</endDate>
22 </period></context>

```

In the data section (listing 5.1), a total of two different context ids are referenced. These are *id210512038457_CY_INSTANT* (lines 1 to 8) and *id210512038457_CY_DURATION* (line 9). The two ids are picked up in the context section (listing 5.2) in lines 1 and 6, respectively, and given a temporal context. *id210512038457_CY_INSTANT* is given a cutoff date of 2020-12-31 (line 4), while *id210512038457_CY_DURATION* is given a period from 2020-01-01 to 2020-12-31 (lines 9 and 10). Both context ids therefore refer to the same fiscal year, but in the data section example, eight values are shown for a cutoff date and one value is shown for the entire period of the year. This is due to business reporting reasons, as some values are related to a point in time, others to a period of time.

In lines 12 and 17, respectively, both types of time context (point in time, time period) are declared again, but in the identifier *CY* is replaced by *PY*. *CY* refers to the current year, while *PY* means the previous year. With these two additional context definitions, it is therefore also possible to publish financial values for the previous year in the report for the current year. This does not replace the previous year's report under any circumstances, but is popular for facilitating the classification of business performance compared to the previous year without having to open the corresponding report.

Next, listing 5.3 gives the unit section for the selected example. Each listed fact in the data section refers to the same unit reference (see listing 5.1, lines 1 through 9). The unit id used (*id210512038457_UNIT*) is resolved in listing 5.3 to the ISO 4217² standard with the currency abbreviation *EUR*. This resolution is required for the later interpretation of

²<https://www.iso.org/iso-4217-currency-codes.html>

the specified values.

Listing 5.3: XBRL file unit section

```
1 <unit id="id210512038457_UNIT"><measure xmlns:iso4217="http://www.xbrl.org/2003/iso4217">iso4217:EUR</measure></unit>
```

Combining the mentioned information, the XBRL file provides for financial entities identified via an id the associated value, the temporal context and the associated unit. The XBRL file can thus be used as a key value dictionary for financial entities of a company. Appendix B.1 contains the complete XBRL file for the example used in extracts here.

5.1.2. XBRL taxonomy

The XBRL taxonomy used in each case has to be specified in the XBRL instance document itself for proper interpretation of the file. Listing 5.4 shows this as an example.

Listing 5.4: XBRL instance document taxonomy reference

```
1 <link:schemaRef xlink:type="simple" xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" xlink:href="http://www.xbrl.de/taxonomies/de-gaap-ci-2016-04-01/de-gaap-ci-2016-04-01-shell.xsd"/>
2 <link:schemaRef xlink:type="simple" xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" xlink:href="http://www.xbrl.de/taxonomies/de-gcd-2016-04-01/de-gcd-2016-04-01-shell.xsd"/>
```

There are two references to two different taxonomies in the given example (line 1 and 2). The attribute *xlink:href* of the *link:schemaRef* xml tag refers to two different taxonomy modules respectively which are linked in the attributes values. The complete taxonomy package is available for download under these links. This makes it possible to completely resolve the facts stated in the XBRL instance file using the referred taxonomies. The use of two different taxonomies is by no means a contradiction. The two taxonomies referenced are de-gcd for meta data such as the company's name and registered office, and de-gaap-ci as a standard taxonomy for financial data that can be applied to a wide range of different companies (see section 2.2.2). Consequently, the taxonomies handle different tasks and complement each other.

5.1.3. XBRL label file

Each taxonomy contains a set of label files to resolve the tags into natural language labels. With the help of domain expertise, tags can sometimes be translated without the

mapping because the tags often use speaking labels. This makes XBRL instances human-readable in places. Nevertheless, each tag refers to a concept, which can be resolved into different languages using the label files. Listing 5.5 shows an excerpt from the latest label file provided with the official taxonomy (Bundesministerium für Finanzen, 2021), which translates the tag *de-gaap-ci_bs.ass.currAss.cashEquiv.bank* into German natural language as an example.

Listing 5.5: XBRL label resolution file extract

```

1 <labelArc xlink:from="de-gaap-ci_bs.ass.currAss.cashEquiv.bank"
2     xlink:to="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_1"
3     xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
4     xlink:type="arc"/>
5 <label xlink:label="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_1"
6     id="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_1"
7     xlink:role="http://www.xbrl.org/2003/role/terseLabel"
8     xlink:type="resource"
9     xml:lang="de">Guthaben bei Kreditinstituten</label>
10 <labelArc xlink:from="de-gaap-ci_bs.ass.currAss.cashEquiv.bank"
11     xlink:to="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_2"
12     xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
13     xlink:type="arc"/>
14 <label xlink:label="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_2"
15     id="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_2"
16     xlink:role="http://www.xbrl.org/2003/role/documentation"
17     xlink:type="resource"
18     xml:lang="de">fuer individuelle Reportingzwecke</label>
19 <labelArc xlink:from="de-gaap-ci_bs.ass.currAss.cashEquiv.bank"
20     xlink:to="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_3"
21     xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
22     xlink:type="arc"/>
23 <label xlink:label="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_3"
24     id="label_de-gaap-ci_bs.ass.currAss.cashEquiv.bank_3"
25     xlink:role="http://www.xbrl.org/2003/role/label"
26     xlink:type="resource"
27     xml:lang="de">Kassenbestand, Bundesbankguthaben, Guthaben bei
    Kreditinstituten und Schecks; Guthaben bei Kreditinstituten</
    label>

```

To resolve an XBRL concept tag, the label file is searched for an exact id match for the *xlink:from* attribute for each *labelArc* element. This is the case three times in the example; in lines 1, 10, 19. With the attribute *xlink:to* all three results refer to a *label* element in the file, which is the next XML element (lines 5, 14, 23). All three results are of type *label* and therefore a result for the search for the concept tag, but they differ in their roles. These are indicated in each case via the attribute *xlink:role* (lines 7, 16, 25). The second result ends in

the attribute *xlink:role* on *documentation*, thus offering additional hints on the use of this concept. In the documentation, references to external sources such as legal texts can also be used, but this is not done in this example. For the pure conversion of the concept tag into natural text, elements with the role to documentation are not relevant and ignored when looking up a label. The other two results in the example are used for this purpose.

The result in line 5 is declared as a short form of the natural text label with the attribute *xlink:role* by the suffix *terseLabel*, while the result in line 23 is declared as a full text translation with the suffix *label*. The short form and the fully spelled out form can each become relevant for different use cases.

Also relevant in each case is the *xml:lang* attribute for determining the language of the natural language label. In the example, the abbreviation for German is specified in all cases, since it is a German label file. Depending on the taxonomy, mapping files for other languages are also provided.

Finally, the natural language form of the concept tag is specified as the value of the XML element. In this case it is "Guthaben bei Kreditinstituten" (bank balances) for the short form. The algorithm to obtain a natural text label for a given XBRL tag id is available on GitHub³

In addition to translating tags into natural language, the label file provides another means of semantic interpretation in XBRL instance documents. The file contains all available concepts identified by the dot-delimited id. From this, a hierarchy of financial entities can be derived. For example, the just translated entity *de-gaap-ci_bs.ass.currAss.cashEquiv.bank* (bank balances) is a sub-entity of *de-gaap-ci_bs.ass.currAss.cashEquiv* (cash-in-hand, central bank balances, bank balances and cheques) and this in turn is a sub-entity of *de-gaap-ci_bs.ass.currAss* (current assets). One step further, the top next upper hierarchy level is reached with *ass* (assets). The remaining prefix of the tag (*de-gaap-ci_bs*) refers to the used German standard taxonomy (*de-gaap-ci*) and to the group of concepts for the balance sheet (*bs*) values.

5.1.4. Text data

The text files of the dataset are in each case the unofficial version published by the Federal Gazette, which includes in full all report components subject to disclosure pursuant to section 325 (1) HGB in one file. The file content is publicly accessible via the website of the Federal Gazette as described in subsection 2.3.1. In terms of content, they correspond to the official disclosure file, but they follow a uniform specification for the format (Bundesanzeiger Verlag GmbH, 2022c). This makes them easier to process than

³https://github.com/farausch/xbrl-distant-supervision/blob/main/entity_resolution.py

company-specific files with individual formatting, etc.

Listing 5.6 shows a highly abbreviated excerpt from one of the text files available for each financial statement. The text files contain a great deal of content that is not relevant to the experiments and to answering the research questions. The excerpt used here is intended only to convey the structure of the data.

Listing 5.6: Report text file extract

```

1 <A>
2   <b>5. Verbindlichkeiten</b>
3 </A>
4 <A>Saemtliche Verbindlichkeiten haben eine Restlaufzeit bis zu einem
   Jahr.</A>
5 <A>Sicherheiten wurden nicht gestellt.</A>
6 <A>Die Verbindlichkeiten aus Steuern betragen EUR 13.370,50 (Vorjahr:
   EUR 9.809,84).</A>
7 <A>Die ausgewiesenen Umsatzerloese (TEUR 388, Vorjahr: TEUR 542)
   betreffen Beratungsleistungen gegenueber verbundenen Unternehmen
   sowie die Leasingertraege. Die sonstigen betrieblichen Ertraege
   enthalten periodenfremde Ertraege von TEUR 3 (Vorjahr: TEUR 8).</A>

```

The text paragraphs required for the experiments are always designated with the XML tag `<A>`. This is the specification of the Federal Gazette for the transmission of the annual financial statement (Bundesanzeiger Verlag GmbH, 2022c) and it is also evident in this example. This uniform representation of natural text allows differentiation from other elements in the document. For example, tables are tagged differently than body text and can therefore be ignored during pre-processing. Nevertheless, text passages tagged with the `<A>` tag must also be heavily filtered in a subsequent step. The example already shows that only a few sections will be of relevance for the experiments. In this case, it is line 6 and 7, as they are the only ones that contain both financial entities and their associated value.

5.1.5. XBRL taxonomy references

The structured storage of the XBRL taxonomy used per XBRL instance document enables an automated evaluation of all different taxonomies used in the dataset at hand. The evaluation is performed to keep the taxonomies for the experiments local rather than downloading them at runtime. Table 5.1 shows the results of this evaluation.

A total of four different versions of the official taxonomy package are applied. From each of the packages the modules `de-gcd` and `de-gaap-ci` are each used together. The oldest version used is from 2007 and the newest from 2016. This circumstance is surprising

Table 5.1.: Reference statistics on used taxonomies in the dataset

Taxonomy module	Amount of references
de-gcd-2007-12-01	604
de-gaap-ci-2007-12-01	604
de-gcd-2010-01-31	3198
de-gaap-ci-2010-01-31	3198
de-gcd-2013-04-30	598
de-gaap-ci-2013-04-30	598
de-gcd-2016-04-01	1204
de-gaap-ci-2016-04-01	1204

insofar as the legislator requires the application of the most current taxonomy in each case (Bundesministerium der Finanzen, 2018). Accordingly, a complete use of all taxonomies from 2007 to 2020 would be expected at this point, since annual financial statements are available in the data set for each year and a new taxonomy has appeared also each year (Rechenzentrum der Finanzverwaltung des Landes Nordrhein-Westfalen (Körperschaft des öffentlichen Rechts), 2021). However, since the data of the Federal Gazette is accepted as correct, the small amount of taxonomies used seem to represent everything necessary for complete and legally valid annual financial statements for large amount of organizations and industries. Accordingly, no quality deficiency or limitation of the dataset can be derived from the statistics. Rather, it shows the broad application possibilities of the standard taxonomies.

5.1.6. XBRL financial entities

Each XBRL instance file contains in the data section a set of key-value pairs for financial entities and their values (see section 5.1.1). The dataset contains references to 412 different financial entities across all XBRL instance files, some of which occur in every instance file based on their frequency, and some of which occur in only one instance file. Table C.1 shows a table with all referenced financial entities and their frequency. Using the four versions of the core taxonomy de-gaap-ci applied, all but eight of the referenced financial entities can be resolved to a plain text label. The entities that cannot be resolved are listed in table C.2 and are not manually translated or similar due to their small number. In addition, these experiments are intended to be built as completely machine repeatable as possible, independent of the input data, so some unresolvable entities are acceptable to achieve the level of automation.

5.2. Approach

The first research question addresses the challenge of annotating the natural text of financial statements algorithmically, without human assistance, using information derived from the XBRL instance files. To achieve this, financial entities and their associated val-

ues must be marked up in the text. The entities and values must also occur together in a sentence. Mentioning a financial entity or the value alone is not sufficient, as the second research question discusses whether a generalized model can be learned that recognizes the entity-to-value relationship. Only from this, key-value pairs can be derived and only with this does the added value of creating a knowledge graph arise, which is the goal of the thesis.

Of all the sentences in the text reports of the financial statements, only a (probably very small) subset is relevant with the given requirements. To filter out the records that contain both a financial entity and its associated numeric value, two consecutive steps are undertaken. First, all records are extracted that contain a numeric value that is also present in the XBRL instance file. Then, in this set of records, the subset is extracted that contains not only the matching numeric value but also the associated financial entity. In case both steps are successful with a specific threshold of certainty the respective tokens of the sentence are annotated. These sentences will subsequently form the data basis for the second research question.

Both numeric values and financial entities can be expressed in a variety of ways due to the flexibility of natural language. Although the label files of the taxonomies provide a plain text label for each financial entity, the same entity can be designated in different ways without incurring a loss of semantics or failing to meet legal requirements. The labels are an aid, not a requirement. There are also linguistic challenges such as abbreviations, synonyms, misspellings, etc. The numerical values can also be labeled in different ways, but a much smaller variety is assumed. Therefore, it is first filtered for sentences with matching numerical values in order to perform the more complex of the two steps on an already pre-filtered set of sentences and thus obtain a better overall result.

Finding and annotating financial entities and financial values within natural language text can be considered an information retrieval task (Singhal et al., 2001). To measure the performance of the algorithmic annotation, 206 sentences are taken from the dataset, tokenized, and the tokens are annotated manually. The common measures precision and recall (Sokolova et al., 2006) for numeric values and entities are reported.

5.2.1. Financial value annotation

Challenge

The numeric financial values in the XBRL instance files follow a consistent data format. They are rational numbers with two decimal places, stored as text in the XML-based XBRL document. The unit used in all cases is the ISO 4217 standard, the base unit of

Euro. Due to the two decimal places, any value can therefore be specified in cents down to the most fine-grained level. A representative example of the representation form is: `<value>1234567.00</value>`. In the text document of the financial statements, this number can be expressed in German in different ways, which makes the mapping from XBRL to text document a challenge. A listing of example expressions shows the problem:

- 1.234,6 Tsd. Euro
- TEUR 1235
- 1,2 Mio. EUR
- EUR 1,2 Millionen
- 1,2 Mio. €
- 0,01 Mrd. €

All of these examples carry less semantics than the sample number from the XBRL file, but are sufficient in the context of the text report and are all applicable. The listing is sorted so that the first example retained the most semantics and the last example retained the least. Now, for the development of a mapping approach, it is particularly important to keep in mind that the author of the report has a duty to inform and should be interested in the comprehensible communication of the company's financial position. For a compromise between readability and information obligation, the first example is therefore preferable to the last example. Nevertheless, the last example cannot be ruled out either, if, for example, for the sake of uniformity, all financial values of a large company are stated in billions of euros and this one value is the only one that carries inappropriately little semantics.

Approach

For mapping purposes, each sentence of an annual financial statement is now checked to see if it contains an indicator for a financial value (€, EUR, Euro, TEUR). This filters out sentences that contain numerical values with a high similarity to an XBRL value that do not refer to a financial entity and would therefore be false positives. In the records with an indicator for a financial value, all numeric values are now compared with all values from the XBRL file and checked to see if the value in the XBRL starts with the same sequence of digits as the value in the text. If the additional condition that the numeric value in the text must have a minimum length is now introduced, the probability of false positive hits becomes less and less likely as the minimum length increases. The algorithm additionally takes into account the problem of rounded values in the text by accepting the value $x - 1$ as match in addition to the actual value x . The first example in the previous listing is also found this way, since the last digit 6 becomes a 5 and thus there

is no longer a contradiction with the XBRL file. The risk of possible further false positives is accepted. Dots and commas are removed from both the text values and the XBRL values before the comparison, since two numerically equal values would otherwise not be recognized as equal due to the different mantissas. Thus, the comparison is reduced to a simple sequence of digits. This approach is particularly promising because it is easy to implement and the minimum length provides an attribute for controlling the false negative/false positive tradeoff.

Evaluation

The annotation of numerical financial values can be considered very well in isolation, because there is no dependence on other steps. True positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are counted. A true positive only occurs if the correct entity is actually recognized. The detection of an incorrect entity is evaluated as a false positive. Tokens without annotation are not a class of their own. Correct predictions of no annotation are therefore not true positives, but true negatives. Thus, the large group of no annotations does not affect precision and recall. Due to the small size of the validation set in relation to the size of the entire dataset, performance is not measured per individual financial entity.

Table 5.2 shows the evaluation results for the annotation of financial values. The algorithm has a passing parameter with the minimum number of digits for a match to control precision and recall. The performance values are therefore given as a function of this variable. The expectation is a high value for precision as the number of minimum digits increases, because the probability of false positives becomes less likely with each additional digit. At the same time, recall will decrease as the number of minimum digits increases, because fewer numerical values in the text meet the minimum length criterion in the first place and are not considered matching candidates (numbers with less digits than minimum length are ignored).

Table 5.2.: Initial performance evaluation - annotation of financial values - correct entity prediction

Minimum number of digits	TP	TN	FP	FN	Precision	Recall
1	169	267	120	8	0.58	0.95
2	169	287	100	8	0.63	0.95
3	157	341	37	29	0.81	0.84
4	107	356	18	83	0.86	0.56
5	57	364	7	136	0.89	0.30
6	24	365	5	170	0.83	0.12
7	14	366	3	181	0.82	0.07

For recall, the expectation is confirmed with the test data, while precision behaves as

expected only up to the minimum length of five digits. The reason for this is the small number of hits that is still delivered with the high value for the minimum length. With the minimum length of 6 digits, the number of hits (true positives) decreases sharply. The number of false positives also decreases, but proportionally not as much. Therefore, the precision decreases and also loses significance due to the small amount of data. Since the recall is below 10% from a minimum length of seven digits and thus hardly delivers any hits, this is the maximum value and the experiment is stopped.

Another noticeable feature is that the recall does not have a 100% recall with a minimum length of only one digit. With the algorithm presented in section 5.2.1, the recall should be exactly 100% in this case. Surprisingly, the examination of the eight false negatives that were not to be expected showed that the values in the text were more finely granular than in the corresponding XBRL file. In these cases, the XBRL file contains rounded values. As a result, the starting sequence of digits in the XBRL is different than in the text and no match is detected. However, the XBRL files refer to the unit Euro with two decimal places in the file unit section (see listing 5.3). Strictly speaking, the values in the XBRL file are therefore wrong in these eight cases, since they do not carry the fully available semantics. Accordingly, the cases need not be considered further.

In the previous evaluation, TP are counted only if the algorithm predicts the correct entity. It is not enough that an entity is detected at all if it is incorrect. In a second evaluation, the measurements are now performed again. However, TP are also counted if the wrong entity is predicted. It is sufficient to detect an entity at all. Table 5.3 shows the results in the same form as before.

Table 5.3.: Initial performance evaluation - annotation of financial values - any entity prediction

Minimum number of digits	TP	TN	FP	FN	Precision	Recall
1	187	267	102	8	0.65	0.96
2	187	287	82	8	0.70	0.96
3	166	341	28	29	0.86	0.85
4	122	356	13	83	0.90	0.57
5	59	364	5	136	0.92	0.30
6	25	365	4	170	0.86	0.13
7	14	366	3	181	0.82	0.07

As expected, the maximum precision and recall is slightly higher with the facilitated requirements for a true positive match. From the motivation of this work presented in the introduction, it is clear that the recognition of XBRL entities and XBRL values is the primary goal. For an automatic graph comparison with the XBRL file, the correct XBRL entity must be recognized as in the first evaluation, however, the recognition of financial

entities and values as key-value pairs alone without a link to the correct XBRL entity also adds value in practice from the auditor's perspective. Therefore, these evaluation results can also be used for subsequent interpretations and decisions in the context of this work.

5.2.2. Financial entity annotation

Challenge

For each XBRL entity, a natural language expression is available in the label file of the taxonomy. In each sentence, these expressions can be searched to find XBRL entities in the text. However, the labels are only one of many ways to uniquely reference a financial entity. Although the flexibility is limited compared to the general flexibility of natural language because there are requirements from the legislator for naming individual financial items [e.g. section 275 HGB, section 266 HGB], these are often abstract and can subsequently be named more specifically for a company. The labels often also refer to entire groups of financial entities, of which only one or a few are actually used. In addition, synonyms pose a further challenge as well, since a financial entity may well be designated in different ways. Consequently, a simple string comparison does not seem a promising approach to find the labels in the text.

Approach

The algorithm for finding and annotating financial entities in text expects as passing parameters a sentence in spaCy data format, a list of expected XBRL entities, a maximum length for forming n-grams, and a minimum similarity threshold value from 0 to 1 to classify hits. The algorithm now creates all possible n-grams with the given maximum length from the sentence. Each of these n-grams is compared to each label of the passed XBRL entities and the similarity is determined in an external function. If the similarity is above the threshold and also greater than the highest similarity so far for the given XBRL entity, then this is saved as a new best match and the tokens of the n-gram are annotated accordingly while the previous annotation for this entity is removed. This implementation thus ensures that the best match is found for each XBRL entity and also that multiple XBRL entities can be found and annotated in one sentence.

Evaluation

The isolated consideration of the annotation results of the textual financial entities can only be used with limitations for goal-oriented interpretation, because the presented algorithm with the passing parameter of the expected entities by design has a dependency on the previous step of the numerical annotations. In the final setup, only those entities are to be passed that are expected because their numeric value was found in the sentence. If, however, the list of all XBRL entities in the corresponding XBRL file is passed for the

parameter, the dependency is resolved and the evaluation can be performed independently of the previous step. But the unanticipated use of the algorithm must be taken into account when interpreting the results because the much larger list of potential candidates might lead to lots of false positives. Performing this evaluation nevertheless is still important to get an indication whether the filtered XBRL entities list is really of such great importance. Without this evaluation, the basis for comparison would be missing.

For the initial evaluation, the static word vectors already available in the spaCy model *de_core_news_lg*⁴ are used to compare strings for similarity and the evaluation is reported for different values of minimum similarity for a match. As a list of expected entities, all entities of the XBRL file are passed as mentioned. Potentially, they can all be present in the report. The algorithm also expects a value for the maximum length of n-grams to be formed for similarity comparison. Here, the value 5 is passed, since this is the maximum value for a contiguous token sequence of a financial entity in the evaluation data. Table 5.4 shows the results depending on the minimum similarity between the n-grams in the text and all labels of the associated XBRL file. As with financial values, TP is initially captured only when the correct entity is recognized, not just any entity at all. The correct prediction of no label for a token is again a TN to obtain meaningful values for precision and recall.

Table 5.4.: Initial performance evaluation - annotation of financial entities - correct entity prediction

Minimum similarity	TP	TN	FP	FN	Precision	Recall
0.1	50	1790	2247	62	0.02	0.45
0.3	50	1874	2163	62	0.02	0.45
0.5	50	2089	1939	71	0.03	0.41
0.7	88	2673	1296	92	0.06	0.49
0.9	162	3468	345	174	0.32	0.48
1.0	128	3701	79	241	0.62	0.35

As expected, precision increases with increasing minimum similarity. However, the maximum precision of 0.62 is not sufficient and there is no room for increasing the minimum similarity further. Such a low precision will not generate reliable training data and thus would not form a useful data basis for the subsequent second research question. A recall of only 35% in this case also carries the risk that the amount of automatically generated training data is no longer large enough for the second research question. However, the amount of the resulting training data for the second research question is not measured at this point and is only a non-negligible point for later. The recall primarily decides the amount of training data, since sentences without hits are discarded for the second part of the experiments.

⁴https://spacy.io/models/de#de_core_news_lg

The reason for the low precision is that only one to a few entities are referenced in each individual sentence. However, as explained at the beginning, all entities of the XBRL file are passed to the algorithm as potential candidates in this evaluation to make it independent from the financial value annotation. This leads to the fact that the algorithm still finds wrong matches even with high requirements for the similarity between text and XBRL label, but these matches are false positives. The large amount of potential entities seems to ensure that a match is likely to be found because of the large selection of candidates but only rarely the correct entity is discovered by that.

Furthermore it is noticeable that the precision is not 100% even with the requirement of an exact match (similarity threshold 1). The reason for this is that different XBRL entities have the same plain text designation. For example, both the entity *de-gaap-ci:incomeuse.gainloss* and *de-gaap-ci:incomeuse.gainloss.netincome* from the HGB standard taxonomy can be expressed with the label text *Jahresueberschuss*, among others. However, the algorithm must decide on a single entity, so the wrong one may be chosen. This is an argument to perform an evaluation in this case as well, like with the financial values before, whether the algorithm recognizes an entity at all, even if it is the wrong one. Table 5.5 shows the results of this evaluation.

Table 5.5.: Initial performance evaluation - annotation of financial entities - any entity prediction

Minimum similarity	TP	TN	FP	FN	Precision	Recall
0.1	315	1790	1982	62	0.14	0.84
0.3	315	1874	1898	62	0.14	0.84
0.5	306	2089	1683	71	0.15	0.81
0.7	285	2673	1099	92	0.21	0.76
0.9	203	3468	304	174	0.40	0.54
1.0	136	3701	71	241	0.66	0.36

The best values for precision and recall have hardly improved significantly, although now the correct entity no longer has to be predicted. It is sufficient to recognize an entity at all. With the interpretation of the previous results where the correct entity had to be predicted, a much higher precision per minimum similarity would now be expected. For the explanation of this little improvement and especially of the precision smaller than 1 for a minimum similarity of 1 (here a precision of 1 would be expected), another property of the evaluation data set is now important. The dataset contains records in which financial entities occur but are not annotated as such. This is because the entity is named in the text, but the value is not identified in the XBRL file. So entities are detected in the text that are purely linguistically an entity, but the algorithm is not supposed to find it because it is not in part of the XBRL file and therefore not relevant. Of relevance are enti-

ties only that are backed by the XBRL file. This is the reason for the still high number of false positives and why the isolated view on only the financial entity annotation has very limited significance. The results are nevertheless a starting point for the improvements that now follow and where passing a filtered set of detectable entities will play a vital role.

5.2.3. Improvement and completion approaches

Both the annotation of financial values and financial entities still have room for improvement. Furthermore, the current implementation does not yet fulfill all requirements in terms of the research question. For example, financial values can be found without a corresponding entity. Apart from that, the results are not yet sufficient. Therefore, improvement and completion measures are implemented in this section.

Filtered entity list

It is debatable whether filtering the potential financial entities as a passing parameter for the algorithm to detect such is an improvement or simply the correct application of the algorithm. Nonetheless, the evaluation for textual financial entity recognition is now being re-run. Since the list of expected entities is now limited to those that were previously recognized numerically, the list is heavily filtered and another variable is added on which precision and recall depend. Precision and recall are now functionally dependent on the minimum number of digits from the numeric entity recognition algorithm and on the minimum similarity from the textual financial entity recognition algorithm. All other parameters remain unchanged from the initial setup. Table 5.6 shows the results of the evaluation. The table is filtered to the sections that are promising for the research question. In particular, low precision is not justifiable, since this is not an indication of reliable annotations and, consequently, high-quality training data cannot be expected. The full table can be found in appendix D.1.

For each minimum similarity value, the precision values have improved significantly compared to the initial setup, because searching for fewer entities leads to fewer false positives. As expected, the recall decreases in each case because not all entities are searched for anymore. The higher the value for the minimum length of a numeric value, the fewer actual matches are found and passed on to the second algorithm. Therefore, as the minimum length of the numerical value increases, the precision tends to increase, while the recall decreases.

Of particular note in this run of the experiment is that very high precision can be achieved for all three minimum lengths shown for the numerical match. Recall decreases with increasing precision, however, sentences without a match for a financial entity and

Table 5.6.: Performance evaluation - annotation of financial entities with filtered expected entity list - correct entity prediction

Min number of digits	Min similarity	TP	TN	FP	FN	Precision	Recall
1	0.7	190	3704	76	179	0.71	0.51
	0.8	178	3731	42	198	0.81	0.47
	0.9	159	3751	21	218	0.88	0.42
	1.0	109	3770	2	268	0.98	0.29
2	0.7	190	3704	76	179	0.71	0.51
	0.8	178	3731	42	198	0.81	0.47
	0.9	159	3751	21	218	0.88	0.42
	1.0	109	3770	2	268	0.98	0.29
3	0.7	172	3727	48	202	0.78	0.46
	0.8	159	3740	33	217	0.83	0.42
	0.9	145	3759	13	232	0.92	0.38
	1.0	99	3770	2	278	0.98	0.26
4	0.7	105	3744	30	270	0.78	0.28
	0.8	97	3750	22	280	0.81	0.26
	0.9	84	3763	9	293	0.90	0.22
	1.0	64	3770	2	313	0.97	0.17

associated value can simply be discarded even if these are false negatives. Thus, low recall simply means less training data. Precision is the much more important value since quality is more important than quantity.

This experiment run also raises the question of how false positives are possible with a minimum similarity of 1, which is equal to an exact string match. There are two of it for each minimum number of digits and they are the same in each case. A manual examination of these cases revealed errors in the annotated data. Because of the small number and to preserve the authenticity of the experiments, the annotated data are not corrected. These errors are human and must be expected and dealt with.

This improvement brings the precision measured from the evaluation data much closer to the goal of a precision of 1 with an acceptable loss of recall. The measure is integrated in the algorithm⁵ and used in the following measures. Since the precision at this point is already very high, the evaluation is not performed again with the facilitated requirement to detect an entity at all. The correct entity must always be predicted for a TP.

Label resolution dictionary customization

The label file is used to obtain the natural language expression of financial entities and then to find them in the text. The applicability of the file for this purpose is therefore of crucial importance for answering the first research question. During the experiments and

⁵https://github.com/farausch/xbrl-distant-supervision/blob/main/text_annotation.py#L25

their manual evaluation, it is noticeable that there are sometimes large linguistic differences between the natural language expression from the label file and the actual reference to the entity in the financial statements. For example, the label file provides for the entity *de-gaap-ci:bs.eqliab.equity.subscribed* the expression *Gezeichnetes Kapital / Kapitalkonto / Kapitalanteile* (eng. *subscribed capital / capital account / capital shares*). Under German law, however, *gezeichnetes Kapital* is reported as *Grundkapital* (eng. *share capital*) [section 152 I AktG] in stock corporations, *Stammkapital* (eng. *share capital*) [section 42 I GmbHG] in a limited liability company and *Geschäftsguthaben* (eng. *business assets*) [section 337 I HGB] in registered cooperatives. Listings 5.7, 5.8 and 5.9 show the resulting low similarity when comparing the vector representations.

Listing 5.7: Low similarity between suggested natural language label and applied expression in the text

```
1 Token span 1: Grundkapital
2 Token span 2: Gezeichnetes Kapital / Kapitalkonto / Kapitalanteile
3 Similarity   : 0.21
```

Listing 5.8: Low similarity between suggested natural language label and applied expression in the text

```
1 Token span 1: Stammkapital
2 Token span 2: Gezeichnetes Kapital / Kapitalkonto / Kapitalanteile
3 Similarity   : 0.21
```

Listing 5.9: Low similarity between suggested natural language label and applied expression in the text

```
1 Token span 1: Geschäftsguthaben
2 Token span 2: Gezeichnetes Kapital / Kapitalkonto / Kapitalanteile
3 Similarity   : 0.10
```

The label for the entity *de-gaap-ci:bs.eqliab.equity.subscribed* reveals yet another problem that applies generally to many of the labels. The labels are very helpful in conveying the linked financial entity to a human, but they are not explicitly used that way in the text. In this example, three different label options are listed in a single string separated by slashes. However, only one of these would be used in the text and in this specific case even none of it explicitly. This gives rise to the hypothesis that by adjusting the label file, the overall evaluation results can be improved because the matches are found more reliably. The aim of improving the labels is to reformulate the list of labels for each financial entity as they would typically be used in a text. For the label *de-gaap-ci:bs.eqliab.equity.subscribed*, for example, the labels *Stammkapital*, *Grundkapital*, *Gezeichnetes Kapital*, *Kapitalkonto*, *Geschäftsguthaben* and *Kapitalanteile* are listed. This is how they

would typically be used in a text and this is exactly how the other entities are treated as well. The entity detection algorithm then calculates the similarity for each of these labels with actual n-grams in the text and searches for matches based on their vector similarity.

This approach involves a lot of manual effort and in certain cases requires comprehensive, domain-specific expert knowledge, as shown in this example. However, since the customized XBRL entity to label file is static and can also be used across many versions of the taxonomies, the effort is initially one-time and the mapping file can be reused in different experiments. Therefore, this approach is considered worthwhile and implemented. Table 5.7 shows the results of the evaluation after customizing the entity label dictionary in comparison to the standard dictionary extracted from the taxonomy used before. Again, the whole table is to be found in appendix D.1.

Table 5.7.: Performance evaluation - annotation of financial entities with filtered expected entity list and customized entity label dictionary - correct entity prediction

Min digits	Min similarity	Precision	Diff	Recall	Diff
1	0.7	0.73	0.06	0.57	0.06
	0.8	0.82	0.01	0.52	0.05
	0.9	0.86	0.03	0.48	0.06
	1.0	0.97	-0.01	0.34	0.05
2	0.7	0.74	0.03	0.60	0.09
	0.8	0.82	0.01	0.52	0.05
	0.9	0.87	-0.01	0.48	0.06
	1.0	0.97	-0.01	0.32	0.03
3	0.7	0.81	0.03	0.50	0.04
	0.8	0.84	0.01	0.46	0.04
	0.9	0.89	-0.03	0.42	0.04
	1.0	0.96	-0.02	0.28	0.02
4	0.7	0.83	0.05	0.33	0.05
	0.8	0.86	0.05	0.32	0.06
	0.9	0.88	-0.02	0.26	0.04
	1.0	0.95	-0.02	0.21	0.04

Recall has improved in all cases shown in comparison to the original entity label dictionary derived from the taxonomy. The customized entity label file provides more true positives in most cases, which is an indication that the customized labels better reflect the actual usage of the entities in the text. In terms of precision, it can be seen that better results are obtained in each case for lower minimum similarity requirements when comparing the word vectors. However, for minimum similarity requirements close to 1, the precision decreases slightly because of some more false positives. The decrease is minimal and does not significantly reduce the overall improvement. The adjusted entity label resolution file is therefore used in subsequent experiments from here on and is also

published in the GitHub repository⁶.

Multi-match barrier

For the extraction of entity-value pairs in the context of the second research question, training data is required where both a financial entity and its associated numeric value are present in each of individual sentences. Multiple financial entities and values may occur within a sentence and the model should predict which tokens form related pairs. Therefore, in this improvement step, the algorithms for recognizing XBRL entities and values in text, which have so far been largely independent of each other, are now linked. Only if both find a matching pair, the tokens are annotated accordingly. Records in which only the financial entity or the value of a financial entity is found are not annotated. The sentences without annotations are subsequently discarded as part of the second research question and are not considered further. Table 5.8 shows the evaluation results for the annotation of financial values after this improvement measure and the whole table is in appendix D.2. The filtered entity list and the customized entity label resolution dictionary from the previous improvement steps are also applied.

As seen in the *Diff* column which refers to the results from the previous improvement implementation, precision has increased almost across the board. This was to be expected because hits for financial values are now discarded if the corresponding financial entity is not also found in the text with the respective minimum similarity. The number of false positives is thus reduced and affects precision as shown here. On the other hand, the recall is reduced because for correctly found financial values the corresponding entity may not be found and thus a true positive is lost. This improvement measure is mandatory in the context of the research question and must therefore be incorporated regardless of the evaluation results. From this perspective, the positive impact on precision is all the more purposeful.

In this evaluation, only the annotation of financial values is discussed, not their entities. At this point, the entity recognition algorithm already receives only the list of entities for which the financial value was found with a certain degree of certainty. This improvement step can therefore have no effect on the recognition of financial entities. It refers only to the financial values.

Different word vectors

To calculate the similarity of two n-grams, the word vectors integrated in spaCy for the language model *de_core_news_lg* have been applied so far. In this evaluation run, these vectors are replaced by the German FastText vectors⁷ (Grave et al., 2018) and the cosine

⁶https://github.com/farausch/xbrl-distant-supervision/blob/main/data/entity_label_dict_customized.json

⁷<https://fasttext.cc/docs/en/crawl-vectors.html>

Table 5.8.: Performance evaluation - annotation of financial values with filtered expected entity list, customized entity label dictionary and multi-match barrier - correct entity prediction

Min digits	Min similarity	Precision	Diff	Recall	Diff
1	0.10	0.64	0.06	0.77	-0.18
	0.30	0.63	0.05	0.77	-0.18
	0.50	0.65	0.07	0.79	-0.16
	0.70	0.75	0.17	0.71	-0.24
	0.90	0.83	0.25	0.57	-0.38
	1.00	0.92	0.34	0.42	-0.53
2	0.10	0.66	0.03	0.79	-0.16
	0.30	0.66	0.03	0.78	-0.17
	0.50	0.67	0.04	0.79	-0.16
	0.70	0.77	0.14	0.72	-0.23
	0.90	0.83	0.20	0.57	-0.38
	1.00	0.92	0.29	0.42	-0.53
3	0.10	0.82	0.01	0.76	-0.08
	0.30	0.82	0.01	0.75	-0.09
	0.50	0.83	0.02	0.74	-0.10
	0.70	0.88	0.06	0.67	-0.17
	0.90	0.89	0.08	0.54	-0.30
	1.00	0.94	0.13	0.40	-0.44
4	0.10	0.86	0.00	0.53	-0.03
	0.30	0.85	-0.01	0.52	-0.04
	0.50	0.87	0.01	0.51	-0.05
	0.70	0.89	0.03	0.45	-0.11
	0.90	0.90	0.04	0.38	-0.19
	1.00	0.93	0.07	0.30	-0.26

distance is measured. 1 minus cosine distance then corresponds to the similarity of two n-grams. Table 5.9 shows the results of the evaluation. The whole table is presented in appendix D.3. All previous improvement measures are kept and the *Diff* columns refer accordingly to the results of the last improvement measure.

For the detection of financial entities, the exchange of vectors has again noticeably increased the precision compared to the already improved results. It is striking that the change in precision for a minimum vector similarity of 1 remains unchanged in each case. This is because the label for the entity and the actual use in the text already had to be exactly the same in the previous run. An increase is therefore no longer possible. Overall, the precision for entity recognition increases significantly with a reasonable decrease in recall. For the recognition of financial values, the change in precision is in each case very close to the corresponding value for the recognition of entities. This is expected, since this value is very strongly dependent now on the recognition of the financial entities due to the multi-match barrier. The recall is pulled down a bit more for the recognition of financial values because some more true positives are lost due to the higher precision

Table 5.9.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using FastText word embeddings - correct entity prediction

Min dig.	Min sim.	Financial values				Financial entities			
		Precis.	Diff	Recall	Diff	Precis.	Diff	Recall	Diff
1	0,70	0,92	0,17	0,62	-0,09	0,89	0,16	0,56	0,00
	0,80	0,93	0,10	0,54	-0,10	0,92	0,10	0,49	-0,04
	0,90	0,92	0,08	0,46	-0,12	0,94	0,07	0,39	-0,09
	1,00	0,91	-0,01	0,42	0,00	0,96	0,00	0,35	0,03
2	0,70	0,92	0,16	0,62	-0,10	0,89	0,15	0,56	-0,01
	0,80	0,93	0,10	0,54	-0,10	0,92	0,10	0,49	-0,04
	0,90	0,92	0,08	0,46	-0,12	0,94	0,07	0,39	-0,09
	1,00	0,91	-0,01	0,42	0,00	0,96	0,00	0,35	0,03
3	0,70	0,93	0,06	0,58	-0,09	0,90	0,09	0,51	0,01
	0,80	0,94	0,05	0,50	-0,10	0,91	0,07	0,43	-0,03
	0,90	0,93	0,04	0,44	-0,10	0,93	0,04	0,35	-0,07
	1,00	0,93	-0,01	0,40	0,00	0,96	0,00	0,31	0,03
4	0,70	0,93	0,03	0,40	-0,04	0,88	0,05	0,31	-0,02
	0,80	0,94	0,04	0,34	-0,08	0,92	0,05	0,26	-0,06
	0,90	0,94	0,04	0,32	-0,06	0,91	0,02	0,23	-0,06
	1,00	0,93	0,00	0,30	0,00	0,95	0,00	0,21	0,00

in the recognition of financial entities. However, this is also within an acceptable range, since precision is more important than recall and the FastText vectors are therefore built into the algorithm^{8,9} as standard from now on.

Non-vector approach

Although word vectors with semantic information about tokens are necessary to also find synonyms in the context of these experiments, an alternative approach is tried and empirically measured in this experiment run. Instead of word vectors, the Levenshtein distance is now used as a comparison method between strings. 1 minus the distance is the similarity in this case and the distance is normalized. The incorporation into the existing algorithm is accordingly simple, because only another method is called for the computation of the similarity. The baseline experiment for the evaluation comparison is the previous experiment run, i.e., table 5.9. The framework conditions are therefore the same in this run with the exception of the similarity measure for financial entities. Table 5.10 shows the results of the evaluation and the complete table is in D.4.

The results show that per minimum number of digits, higher precision can be achieved for smaller values of the minimum similarity measure. Values of this range are achieved with FastText vectors only later (higher similarity threshold). However, this is only due

⁸<https://github.com/farausch/xbrl-distant-supervision/blob/main/config.py#L24>

⁹https://github.com/farausch/xbrl-distant-supervision/blob/main/similarity_measure.py

Table 5.10.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using Levenshtein distance as similarity measure - correct entity prediction

Min dig.	Min sim.	Financial values				Financial entities			
		Precis.	Diff	Recall	Diff	Precis.	Diff	Recall	Diff
1	0.5	0.90	0.26	0.67	-0.10	0.87	0.46	0.59	-0.04
	0.6	0.92	0.15	0.63	-0.06	0.91	0.33	0.57	-0.04
	0.7	0.93	0.01	0.59	-0.02	0.92	0.04	0.53	-0.03
	0.8	0.93	0.00	0.58	0.04	0.94	0.02	0.52	0.03
	0.9	0.93	0.01	0.56	0.10	0.96	0.02	0.50	0.11
2	0.5	0.90	0.20	0.67	-0.13	0.87	0.42	0.59	-0.05
	0.6	0.92	0.10	0.63	-0.08	0.91	0.27	0.57	-0.04
	0.7	0.93	0.01	0.59	-0.02	0.92	0.04	0.53	-0.03
	0.8	0.93	0.00	0.58	0.04	0.94	0.02	0.52	0.03
	0.9	0.93	0.01	0.56	0.10	0.96	0.02	0.50	0.11
3	0.5	0.91	0.08	0.62	-0.14	0.87	0.32	0.53	-0.05
	0.6	0.93	0.02	0.59	-0.08	0.91	0.19	0.51	-0.04
	0.7	0.94	0.01	0.56	-0.02	0.92	0.02	0.48	-0.02
	0.8	0.94	0.00	0.54	0.04	0.94	0.02	0.47	0.04
	0.9	0.94	0.00	0.53	0.09	0.96	0.03	0.45	0.10
4	0.5	0.92	0.05	0.43	-0.09	0.87	0.32	0.34	-0.01
	0.6	0.93	0.00	0.41	-0.06	0.92	0.18	0.33	-0.01
	0.7	0.94	0.01	0.39	-0.01	0.93	0.05	0.31	-0.01
	0.8	0.94	-0.01	0.38	0.05	0.93	0.02	0.30	0.05
	0.9	0.94	0.00	0.37	0.06	0.96	0.06	0.29	0.06

to the different types of the similarity measure. Therefore, very high deviations are especially noticeable for minimum similarities up to 0.6. Above a similarity threshold, only small differences can be measured for the Levenshtein distance instead of the FastText vectors, although a slight improvement can be seen in the tendency of the precision for both financial entities and financial values. There is no difference for each minimum digit length with a similarity threshold of 1.0 which is why this row is missing in this presentation.

The evaluation results are interesting in that better results were expected with semantic word vectors than with a sequence-based similarity measure. The Levenshtein approach does not recognize synonyms and word substitutions are not compensated for. Explanations for this can only be given with caution because the evaluation data were not analyzed sentence by sentence. For example, it could be that hardly any synonyms were used in the evaluation data, hardly any word substitutions occur, and most entities in the text are actually referenced very closely to the wording of the label specifications. However, it is also possible that synonyms are not recognized by the word vectors either and therefore have no positive effect on the measurement results compared to Levenshtein.

The vocabulary in these experiments is domain specific and therefore not necessarily reflected in the full semantics in the vectors.

5.3. Evaluation

Finishing now is the final evaluation with all improvement measures built in. The list of potential entities for financial entity detection is filtered, the entity label dictionary is manually adjusted, the multi-match barrier is enabled, and for the similarity calculation both the FastText vectors with cosine similarity and the Levenshtein distance are used since the evaluation results are very similar. Precision and recall are again reported as a function of the minimum number of digits a numeric token must have to even be considered a financial value and the minimum similarity between the financial entity label and its actual use in the text as an n-gram. Precision and recall are now no longer reported separately for financial entity and financial value detection, but as the average of the two. An equally weighted mean is used here because the annotation of the financial values would otherwise be severely underrepresented due to their shorter sequence length which results in less tokens annotated as financial values compared to their associated entities. Financial entities consist of more tokens than financial values. For meaningful evaluation values in the sense of the research question, however, these should be equally weighted, since the annotation of financial values is just as important as that of financial entities, regardless of their individual sequence length. Finally, the Fbeta value is calculated with a beta of 0.25 as the harmonized mean of the two. This weighting places much more emphasis on precision than on recall. High quality annotations are more important than many annotations. Table 5.11 shows the relevant results using FastText vectors as similarity measure, the entire table is in appendix D.5.

Table 5.12 show the final evaluation results using Levenshtein distance instead of FastText vectors and the whole table is in appendix D.6.

Across both evaluations, the best overall score is a value for F0.25 of 0.9. The value is obtained using Levenshtein distance as a similarity measure several times, for a minimum number of digits 1, 2, and 3, respectively. With a value for F0.25 of 0.88, performance with FastText vectors is only 2 percent behind overall. Here, the best value is also achieved several times, also for minimum numbers for digits of 1, 2 and 3. Both approaches share this property. The maximum precision achieved is 0.95 overall, again with the Levenshtein approach. The maximum precision with FastText vectors is not far behind with 0.94, but this value is only achieved with exactly the same strings (similarity of 1). This is not the intention when using the vectors because this equals a character-based string comparison. The best values for precision are thus close to each other, but the Levenshtein approach can maintain a higher recall in all cases. For a precision of 0.95

Table 5.11.: Final performance evaluation - annotation of financial values and financial entities using FastText vector comparison

Min digits	Min similarity	Precision	Recall	F0.25
1	0.70	0.90	0.59	0.88
	0.80	0.92	0.52	0.88
	0.90	0.93	0.42	0.87
	1.00	0.94	0.39	0.86
2	0.70	0.90	0.59	0.88
	0.80	0.92	0.52	0.88
	0.90	0.93	0.42	0.87
	1.00	0.94	0.39	0.86
3	0.70	0.91	0.54	0.88
	0.80	0.93	0.47	0.88
	0.90	0.93	0.40	0.86
	1.00	0.94	0.36	0.86
4	0.70	0.90	0.36	0.83
	0.80	0.93	0.30	0.83
	0.90	0.92	0.27	0.81
	1.00	0.94	0.25	0.81

Table 5.12.: Final performance evaluation - annotation of financial values and financial entities using Levenshtein distance

Min digits	Min similarity	Precision	Recall	F0.25
1	0.7	0.93	0.56	0.89
	0.8	0.93	0.55	0.90
	0.9	0.95	0.53	0.90
	1.0	0.94	0.39	0.86
2	0.7	0.93	0.56	0.89
	0.8	0.93	0.55	0.90
	0.9	0.95	0.53	0.90
	1.0	0.94	0.39	0.86
3	0.7	0.93	0.52	0.89
	0.8	0.94	0.50	0.89
	0.9	0.95	0.49	0.90
	1.0	0.94	0.36	0.86
4	0.7	0.93	0.35	0.85
	0.8	0.94	0.34	0.85
	0.9	0.95	0.33	0.86
	1.0	0.94	0.25	0.81

a recall of more than 0.5 is still possible, for a precision of 0.94 with the FastText vectors only a maximum recall of 0.39. The evaluation script is available on GitHub¹⁰.

5.4. Distant supervision algorithm output examples and analysis

The annotated output sentences from the distant supervision algorithm can be classified into three categories. Ideally, in a sentence a) all financial entities and financial values are recognized and correctly linked by the specific XBRL annotation. Otherwise b) it is possible that not all entities and values are recognized within a sentence. Once a single entity with value is recognized, the sentence is part of the training dataset. Here the problem of low recall becomes noticeable. The third category c) are wrong annotations. Both entities and their associated values can be incorrectly annotated. Here, the imperfect precision of the annotation algorithm becomes noticeable. This section provides an example for each of these categories.

The first example in listing 5.10 shows an output sentence that specifies a financial entity, the associated value for the current and previous years, and the relative change. Here, the annotation algorithm has correctly marked the financial entity and the financial value for the current fiscal year. The selected XBRL entity for material cost is also correct.

Listing 5.10: Example 1 from the algorithmically generated dataset

```
1 financial-reports/batch1/XBRL/190814021079.xbrl
2 Der;Materialaufwand;erhoehte;sich;von;554,1;Mio.;Euro;auf;599,6;Mio.;
   Euro;um;8,2;Prozent;.
3 0;de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.
   materialservices;0;0;0;0;0;0;0;de-gaap-ci:is.netincome.regular.
   operatingtc.grosstradingprofit.materialservices;0;0;0;0;0;0
```

The second example in listing 5.11 shows that correct annotation is possible even in records with multiple financial entities and financial values. The correct tokens were annotated for the inventories and the receivables. However, it should be noted that the receivables are shown as *Forderungen und Sonstigen Vermoegensgegenstaende* in the text, but only the first token *Forderungen* is marked with the correct XBRL entity. There are several labels for the XBRL entity *de-gaap-ci:bs.ass.currass.receiv*, including *Forderungen* and *Forderungen und sonstige Vermoegensgegenstaende*. At first glance, the full n-gram *Forderungen und Sonstigen Vermoegensgegenstaende* seems the better match, however *Forderungen* is a perfect match with no edit costs and the edit cost for *Forderungen und Sonstigen Vermoegensgegenstaende* is 1 because a single letter must be substituted. The reason for this is that the object in this sentence is correctly used in the accusative case, one of the four

¹⁰<https://github.com/farausch/xbrl-distant-supervision/blob/main/evaluation.py>

cases for the relationship of subject to object in German. This leads to the appended *n* in the token *Sonstigen*. This problem would in principle be avoidable with stemming or lemmatizing, however the result shown is not incorrect and the annotation algorithm is already resource intensive. The low recall becomes noticeable within a single entity in this particular case.

Listing 5.11: Example 2 from the algorithmically generated dataset

```

1 financial-reports/batch1/XBRL/111212060744.xbrl
2 Die;Vermögensstruktur;wurde;im;Geschaeftsjahr;2010;ueberwiegend;durch;
  die;Erhoehung;der;Vorraete;um;4.202;TEUR;auf;51.961;TEUR;sowie;der;
  Forderungen;und;Sonstigen;Vermögensgegenstaende;um;4.704;TEUR;auf
  ;59.431;TEUR;gepraegt;.
3 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;de-gaap-ci:bs.ass.currass.inventory;0;0;0;0;de-
  gaap-ci:bs.ass.currass.inventory;0;0;0;0;de-gaap-ci:bs.ass.currass.
  receiv;0;0;0;0;0;0;0;0;0;0;de-gaap-ci:bs.ass.currass.receiv;0;0;0

```

The third example now shows a peculiarity that might turn out to be a problem in the course of the following experiments. In the sentence, the financial entities long-term bank liabilities, provisions and equity are mentioned and and shown with financial values. Thus, three pairs would be expected here, but the algorithm finds only one entity with associated value with the configuration chosen. If the algorithm had not found any of the three entities, the sentence would have been discarded and there would be no problem except one training example less. In this case, however, existing entities and values are not annotated in the training data, but they should be. A statistical model learns false correlations from this. This example shows that the low recall not only eliminates entire sentences (which is fine), but also leaves out annotations within a sentence (which is problematic). This problem will be considered further in the data augmentation section 5.5.2.

Listing 5.12: Example 3 from the algorithmically generated dataset

```

1 financial-reports/batch2/XBRL/111212060744.xbrl
2 Die;Kapitalstruktur;wird;im;Wesentlichen;durch;eine;weitere;Reduzierung
  ;der;langfristigen;Bankverbindlichkeiten;um;3.000;TEUR;auf;0;TEUR
  ;,;einer;Erhoehung;der;Rueckstellungen;um;3.397;TEUR;auf;31.717;
  TEUR;sowie;die;Erhoehung;des;Eigenkapitals;um;1.771;TEUR;auf
  ;110.668;TEUR;beeinflusst;.
3 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;de-gaap-ci:bs.eqliab.
  accruals;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;de-gaap-ci:bs.eqliab.accruals
  ;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

```


sentences per document might well be expected, but the recall is not high and many companies, especially small and medium-sized enterprises, do not deal with the company's financial position in text form. They often leave it at tables. In the 5209 sentences there are 11244 annotated financial entities/values and 5836 relation annotations (meaning two annotations refer to the same entity in a single sentence). There are only 627 relations more than there are sentences at all. This means that a maximum of 627 sentences contain more than one entity-value pair annotation. This does not make the problem of multiple different entities in one sentence (see 5.12) an edge case because as demonstrated there might be false negatives and the actual number is assumed to be higher than 627.

In the training dataset there are 113 different XBRL entities referred to. Subscribed capital is by far the most frequently used (2511 times), with all other XBRL entities being reported less than 600 times. Only 39 entities are mentioned and given value more than 100 times. The dataset is thus relatively unbalanced and compared to the 412 XBRL entities provided in the taxonomy, only slightly more than 1/4 are used. The reason for this could be, on the one hand, that the entities not mentioned are very special and only occur in exceptional cases. After all, the task of the taxonomy is to cover a very wide range of industries. On the other hand, it could also be that the missing entities were mentioned in the text, but were not recognized by the algorithm.

5.5.2. Data augmentation

As shown in the analysis of the dataset, 299 of a total of 418 entities provided by the XBRL taxonomy are missing in the training data because only 113 are present. However, the goal in developing the model is to cover the entire taxonomy if possible and thus to develop a generalized model that recognizes all of the XBRL entities.

Data augmentation deals with diversifying and expanding data collections without explicitly collecting new data (Feng et al., 2021). In this specific case, 5209 records are already available, in each of which it is known where a financial entity is located and which token represents the associated value. In addition, the taxonomy contains a label dictionary that provides the plain text expressions for all possible XBRL entities. In the training sentences, the existing expressions can thus be replaced by other expressions. In this way, training sentences are created that have the same sentence structure as those actually used in the annual financial statements, but with different entities. Listing 5.14 shows an example of an original sentence, and listing 5.15 shows one created artificially with a different entity.

Listing 5.14: Original sentence from the algorithmically generated dataset

```
1 Das;Beteiligungsergebnis;stieg;um;7,8;Mio.;EUR;auf;18,3;Mio.;EUR;.
```

```

2 0;de-gaap-ci:is.netincome.regular.fin.netparticipation;0;0;0;0;0;0;de-
  gaap-ci:is.netincome.regular.fin.netparticipation;0;0;0

```

Listing 5.15: Augmented sentence from the algorithmically generated dataset

```

1 Das;Aufwendungen;fuer;Altersversorgung;stieg;um;7,8;Mio.;EUR;auf;18,3;
  Mio.;EUR;.
2 0;de-gaap-ci:is.netincome.regular.operatingtc.staff.social.socexp;de-
  gaap-ci:is.netincome.regular.operatingtc.staff.social.socexp;de-
  gaap-ci:is.netincome.regular.operatingtc.staff.social.socexp
  ;0;0;0;0;0;0;de-gaap-ci:is.netincome.regular.operatingtc.staff.
  social.socexp;0;0;0

```

In this example, the XBRL entity for the investment result has been replaced by the pension expense. Algorithmically, it is possible to perform any number of these replacements and thus extend the dataset with all missing financial entities. However, a linguistic problem with this approach is already apparent. The first token of the sentence *das* is grammatically incorrect and should correctly read *die* after the following noun has been replaced. The German language distinguishes between three definite articles: *der*, *die* and *das*. However, articles occur so frequently in the German language that their importance for contextual word vectors is negligible and this problem is not pursued further. Also the verb *stieg* after the entity expression would have to be correctly transformed into the plural form *stiegen*. However, since the verb still refers to the same correct infinitive form, this problem is also not to be pursued further.

Another problem with data augmentation originates from the low recall of the annotation algorithm. An entity consisting of several tokens may not be annotated completely, but only single tokens of the correct token sequence. Listing 5.16 shows an example where this is the case. The correct, complete financial entity in this case would be *Forderungen aus Lieferungen und Leistungen* (accounts receivable), but the algorithm has only annotated *Forderungen*. If the entity is now replaced, not the entire token sequence, but only the one word is replaced and *aus Lieferungen und Leistungen* remains without annotation. This not only results in false negatives in the training data, but also in unnatural token sequences, as listing 5.17 shows. The entity trade receivables has been replaced by *Fertige Erzeugnisse* (finished goods). Behind the expression for finished goods, however, the remainder of the unrecognized original expression now remains. The false negatives are unfortunately unavoidable with the annotation algorithm used and the parameters set.

Listing 5.16: Original sentence from the algorithmically generated dataset with false negative annotations

```

1 Der;Grossteil;des;Vermögens;besteht;mit;4,3;Mio.;EUR;(Vj.;0,8;Mio.;
  EUR);aus;Forderungen;aus;Lieferungen;und;Leistungen;.
2 0;0;0;0;0;0;de-gaap-ci:bs.ass.currass.receive;0;0;0;0;0;0;0;0;0;0;de-
  gaap-ci:bs.ass.currass.receive;0;0;0;0;0

```

Listing 5.17: Augmented sentence from the algorithmically generated dataset with false negative annotations in the original one

```

1 Der;Grossteil;des;Vermögens;besteht;mit;4,3;Mio.;EUR;(Vj.;0,8;Mio.;
  EUR);aus;Fertige;Erzeugnisse;aus;Lieferungen;und;Leistungen;.
2 0;0;0;0;0;0;de-gaap-ci:bs.ass.currass.inventory.finishedandmerch.
  merchandise;0;0;0;0;0;0;0;0;0;0;de-gaap-ci:bs.ass.currass.inventory
  .finishedandmerch.merchandise;de-gaap-ci:bs.ass.currass.inventory.
  finishedandmerch.merchandise;0;0;0;0;0

```

The data augmentation is now performed with the existing 5209 sentences. For each sentence where a single entity has been annotated, 4 more sentence are created with different financial entities and remaining financial value. For each sentence in which more than one entity has been annotated, 7 more sentences are created. This results in a total of 59552 sentences in the training dataset. Parameters 4 and 7 are basically customizable. They were chosen so that each XBRL entity is present at least 100 times in the dataset. In addition, this way the relative proportion of sentences in which multiple entities were annotated is artificially increased compared to sentences in which only one entity was found. Thus, the problem of sentences in which some entities were not found at all (see listing 5.12) due to low recall becomes relatively smaller. In other words, the relative proportion of false negatives becomes smaller, since sentences with more entities expectedly have few false negatives, and it is precisely these sentences that are replicated.

With this approach, a) the number of different entities can be increased from 113 to 404 (all XBRL entities for which at least one label exists), b) the problem of false negatives is reduced, because primarily records with many entities are replicated, c) the number of training records is increased from 5209 to 59552 and d) there are at least 100 examples per XBRL entity so that a model can learn from various instances. The script for data augmentation is available on GitHub¹¹.

¹¹https://github.com/farausch/xbrl-distant-supervision/blob/main/data_augmentation.py

6. Financial entity detection and relation extraction

After introducing a method for extracting entities and relations in chapter 4 and a method for automated annotation of German financial statements in chapter 5, this chapter links the two. The automatically generated dataset is used to train the joint entity detection and relation extraction model and to evaluate it afterwards with a separate dataset. This chapter will provide the answer to the second research question.

6.1. Datasets

For training the joint entity detection and relation model, three datasets with different origins and properties are used. For the evaluation, the same test dataset is used in each experiment to ensure comparability. In order to use the datasets presented with the modified SpERT model, either the data input reader of the SpERT code¹ must be adapted or the data must be converted to the CoNLL04 format. The second approach involves less work and CoNLL04 is the better known data structure. Therefore, this approach is followed and the associated script is on GitHub² for reusability.

6.1.1. Automatically annotated training dataset

The first dataset (dataset A, where A stands for algorithmically created) corresponds to the automatically generated dataset from section 5.5.1. It was not augmented in this case, but consists of the original 5209 sentences with the problems explained. This dataset must not be published because the data from the Federal Gazette was only allowed for processing, not publishing, in this work.

6.1.2. Automatically annotated and augmented training dataset

The second dataset (dataset A, A where the first A stands for algorithmically created and the second A stands for augmented) is automatically generated and augmented as explained in section 5.5.2. It is thus created from 5209 sentences augmented to a total of 59552 sentences and addresses some of the problems of the non-augmented dataset. This

¹https://github.com/farausch/spert/blob/master/spert/input_reader.py

²https://github.com/farausch/xbml-distant-supervision/blob/main/conll_transformer.py

dataset must not be published because the data from the Federal Gazette is only allowed for processing, not publishing, in this work.

6.1.3. Manually annotated training dataset

To see the distant supervision approach for automatic annotation of the data compared to manual annotation, in addition to the algorithmically created datasets, one is also created manually and augmented to 7287 sentences (dataset M, A where M stands for manually created and A stands for augmented). For this purpose, 200 randomly selected sentences were chosen from financial statements and manually annotated. Most of these sentences contain at least one pair of financial entity and financial value like the automatically created datasets. The augmented dataset is available on Huggingface³ in the same structure as the CoNLL04 dataset. The original 200 sentences dataset from which the augmented dataset was created is also available on Huggingface⁴.

6.1.4. Test dataset

The test dataset, which is used to evaluate the trained model, consists of 200 manually annotated sentences, also randomly sampled from the financial reports. Most of them contain at least one pair of financial entity and financial value and have an empty intersection with the presented manually annotated training dataset. The dataset is available on Huggingface⁵.

6.2. Approach

The approach used to extract the financial entities and financial values is the modified SpERT model from Section 4. Two different models are used as the pre-trained BERT language model. First, the bert-base-german-cased model⁶ is used, which is provided by the Bavarian State Library and was trained with a total of 16 GB of data. Gururangan et al. (2020) have shown that using a language model refined for the application domain can increase performance. Therefore, the bert-base-german-cased model is additionally refined with 100,000 sentences from financial reports. 50,000 of these sentences were taken unfiltered and randomly from 5,500 financial statement reports and the second 50,000 sentences are of the same origin, but these sentences are filtered to have an indicator of a reference to a financial value. Indicators for a financial value are natural

³https://huggingface.co/datasets/fabianrausch/financial-entities-values-augmented/blob/main/financial_sentences_augmented_from_200.json

⁴https://huggingface.co/datasets/fabianrausch/financial-entities-values-augmented/blob/main/financial_sentences_200.txt

⁵https://huggingface.co/datasets/fabianrausch/financial-entities-values-augmented/blob/main/financial_sentences_200_test.json

⁶<https://huggingface.co/dbmdz/bert-base-german-cased>

language expressions of currencies (e.g. EUR). Huggingface provides a script⁷ for fine-tuning BERT language models, which was used for this purpose. The fine-tuned model `german-financial-statements-bert`⁸ is also open-source. In the evaluation of the experiments, the difference between the models is measured below.

6.3. Evaluation

The evaluation is conducted with a total of five different experiments. The task of the model in each case is to recognize financial entities and financial values (two entity types plus no entity) in the sentences and to predict the correct relationship between them (one relation type plus no relation).

With the original SpERT and the modified SpERT model, two different models are used for evaluation. Modified in this case refers to the added POS and dependency tags on token-level. The shortest dependency path was not applied. The training datasets are the presented data and the BERT model is either `bert-base-german-cased` (GC) or the refined `german-financial-statements-bert` (FS). Precision, recall and F1 are measured separately for entity detection and relation extraction. For a true positive in entity detection, both the correct class and the correct span must be predicted with the correct bounds. For a true positive in relation extraction, both entities must be correct according to the above definition and, in addition, the relation must be predicted in the correct direction and class. The F1 score in relation extraction is therefore the most meaningful value of this evaluation, since it combines most values and requirements. Table 6.1 shows the evaluation, where the first row per experiment shows micro-average values and the second row shows the macro-average values. For relation extraction, there is no distinction between micro- and macro-average, since there is only one relation type plus no relation.

The hyperparameters were not further modified from the original paper except for the number of learning epochs. Optimizer, learning rate schedule, dropout layers etc. remain the same. However, for the best F1 relation extraction result, only 3 epochs were needed in the first two experiments with the algorithmically generated dataset. For the following three experiments with the manually generated dataset, only 2 epochs are needed, after which the model starts to overfit. With the automatically generated dataset, presumably this few iterations are needed because one epoch contains a lot of training data. With the manually generated and augmented data set, even one epoch less is needed. Here, there is the additional characteristic that syntactically there are only 200 different sentences, which have been augmented to about 7000. Within one epoch, the model is therefore already confronted with many similar sentences, which often only differ in the entity. This

⁷https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_mlm.py

⁸<https://huggingface.co/fabianrausch/german-financial-statements-bert>

Table 6.1.: Joint financial entity and value extraction evaluation

Model	Train set	BERT	Entity detection			Relation extraction		
			P	R	F	P	R	F
SpERT (unmodified)	A	GC	87.29 87.11	73.39 75.09	79.74 79.79	62.56	62.01	62.28
SpERT (unmodified)	A, A	GC	89.70 89.49	71.89 73.43	79.87 80.28	71.36	64.19	67.59
SpERT (unmodified)	M, A	GC	83.99 84.42	89.29 90.38	86.56 87.27	75.00	82.76	78.69
SpERT (unmodified)	M, A	FS	86.45 87.18	90.25 91.28	88.31 89.18	78.10	81.47	79.75
SpERT (modified)	M, A	FS	87.88 88.37	88.72 89.92	88.30 89.10	81.20	81.90	81.55

circumstance comes close to having additional epochs. 2 epochs are sufficient for the best result without overfitting setting in.

If the table is read from top to bottom with focus on the relation F1 score, each experiment improves the results of the previous one. With the dataset created algorithmically, by distant supervision and without human annotations, an F1 score of over 62 is already achieved (experiment and row 1). If this experiment is modified by augmenting the data (experiment and line 2), this score is improved by over 5 to just under 68. With the change to the manually created and augmented dataset (experiment and line 3), this value is raised to just under 79, which corresponds to a significant increase. It is particularly noticeable that the precision in the recognition of entities decreases noticeably, while the recall increases significantly at the same time. On the one hand, the high precision of the annotation algorithm is noticeable, and on the other hand, the already discussed problem of low recall. In the next step (experiment and line 4), the manually created and augmented dataset is used again, but the refined BERT model is used. The precision of the entity detection is noticeably improved. Measured by the F1 score of the relation extraction, however, the use of the refined model provides only a minor improvement. The last experiment (experiment and row 5) uses the modified version instead of the original SpERT model and improves the relation extraction F1 score again by about 2 to a final 81.55.

7. Discussion

In this work, three groups of experiments were conducted to answer the two research questions. Chapter 4 presents an adapted approach to joint entity detection and relation extraction, chapter 5 is devoted to devising a method for automatically annotating financial reports using distant supervision, and chapter 6 brings these two together to extract financial entities and financial values from natural language texts.

The adapted SpERT model for joint entity detection and relation extraction with part-of-speech tags and dependency tags from chapter 4 achieves a F1 relation extraction score of 73.15 (micro average) and 73.89 (macro average), respectively. The results correspond to an absolute increase of around 2 percent. On the relation extraction benchmark for the CoNLL04 dataset on *Papers With Code*¹ only the model introduced by Wang and Lu (2020) performs slightly better without additional training data. For other datasets, however, the comparison may look different, as the state-of-the-art results are very close. The use of explicitly encoded linguistic information derived from language parsers has been demonstrated before with slight improvements over the initial model without explicit information and is therefore not new at this point (Santosh et al., 2021). The use of the shortest dependency path (SDP) could not achieve the hoped-for performance increase. In this case, all tokens outside the SDP were ignored during relation classification using an attention mask². The low precision of this approach may indicate that the context consisting only of the tokens of the SDP is not comprehensive enough to be accurately predicted by the model. However, this contradicts the robust findings of Bunescu and Mooney (2005), which elaborated the SDP as a central feature for the classification of relations. Furthermore, despite extensive debugging and manual testing, an implementation error cannot be ruled out, of course. To further improve the results, the model could be extended to consider the distance of the two entity candidates when classifying relations. Lee et al. (2017) have shown that this information is useful for coreference resolution and can also be promising for relation extraction.

Chapter 6 presents the financial entity detection and relation extraction approach with the distant supervision and human annotated dataset. The final result of a F1 relation extraction score of 81.55 is, with respect to the motivation of this work, at least a great help for auditors in their work. The derivation of a knowledge graph from natural text

¹<https://paperswithcode.com/sota/relation-extraction-on-conll04>

²<https://github.com/farausch/spert/blob/feature/dep-tree-shortest-path/spert/sampling.py#L44>

is possible with the test dataset to the measured extent. The improvement measures undertaken were able to contribute to the improvement of performance in the evaluation. The data augmentation, the modifications to the SpERT model, and the fine-tuned BERT language model made an important contribution to the final result.

The separate evaluation files for both entity detection and relation extraction are available on GitHub³ and can be opened with a web browser. A manual error analysis of the entity detection reveals that the most typical mistake in the model predictions is the incorrect span prediction, meaning there are tokens missing or too many tokens marked as entity. This type of error is acceptable in many practical situations because the entity is not completely wrong. The most common error in relation extraction, unsurprisingly, is also the wrong entity span boundary prediction since relation extraction can only be correct if the entity detection was correct beforehand. On top of that, some relations are not properly revealed if the context between two entities consists of only one token which might be too little information. Some sentences of the test data set also contain a financial entity and a financial value, but they do not belong together. Here, the model partially outputs a false positive relation match. This circumstance indicates that more sentences should be used in the training data set where financial entity and value do not belong together. The focus in the training data set is so far clearly on positive examples and the model might not have the possibility to learn these difficult cases because of too few negative examples.

A manual examination of the refined BERT model using token masking shows that it is very biased. For example, if the sentence *The [MASK] beträgt TEUR 100.* is passed to the model, the model returns *Eigenkapital* (eng. equity) with a confidence of 0.99 for the masked token, which almost completely neglects all other possible entities. Equity is one of the most frequently reported values in the text reports, which is why the model learned this prediction. Perhaps it would improve overall performance to also artificially adjust the distribution of financial entities when fine-tuning the BERT model, similar to data augmentation, so that all entities are evenly represented.

The manually annotated, augmented dataset produced the best results after only 2 epochs, before overfitting set in. It might make sense to limit the data augmentation a bit so that a smaller dataset with more epochs is used for training. Forming over 5000 sentences from a dataset with originally only 200 sentences may be a bit too much, since the number of different formulations is still only 200. The motivation was to have many formulation possibilities per entity for the model to learn from. Nevertheless, an attempt with less data seems reasonable to conduct. The CoNLL04 dataset, which was used in chapter 4, also consists of only about 1,000 training sentences while having more entity

³<https://github.com/farausch/spert/tree/master/evaluation>

and relation types to learn.

Regarding the automatic creation of a training dataset using distant supervision in chapter 5, there is much more room for improvement compared to the other two. The best relation extraction F1 score with the algorithmically generated and augmented dataset in the experiments in chapter 6 is 67.59, which is about 14 behind the dataset consisting of only 200 manually annotated and augmented sentences. Annotating 200 sentences has been significantly less effort than developing the method for XBRL distant supervision. Nevertheless, the manual annotation is the significantly better performing approach. The necessity of this little manual training data is an example of the strength of pre-trained language models. One reason for the unsatisfactory results with the distant supervision approach is probably that the presented method was developed with too much focus on precision and too little on recall. For the evaluation, the F0.25 score was used instead of F1 score. The assumption that a low recall is acceptable because the sentences then do not become part of the training dataset has proven to be untenable. If there is more than one entity in a sentence, the low recall often leads to the distant supervision algorithm finding only one of the entities in the sentence (intra-sentence low recall). Even within a single entity consisting of multiple tokens this has proven to be a problem (intra-entity low recall) as discussed in the data augmentation section. This diminishes the quality of the training data and has also been shown in the evaluation of the financial entity detection and relation extraction model. However, simply increasing recall by e.g. lowering the similarity measure will only come at the expense of precision, so that an overall improvement must be questioned. It makes sense to pursue further approaches to increase the recall that result in a reasonable amount of loss of precision. The precision and recall was not measured per XBRL entity type during the evaluation of the distant supervision annotation. The distinction is done on a higher level with financial entities and financial values only. Measuring the evaluation values per XBRL entity might reveal more insights which entities are difficult to annotate and which properties they share and help to derive countermeasures. Moreover it is surprising to see that the Levenshtein similarity measure outperforms the word vector approaches. Using contextualized word embeddings for the entity detection task might be worth trying in order to increase the recall. But it is a specific challenge to generate contextualized word embeddings for the comparison labels since they do not have any/very little context. In addition, recall could be increased by adjusting and systematically evaluating the parameters for the data augmentation algorithm. In the experiments of this work, each sentence with one entity is augmented by 4 more sentences and each sentence with more than one entity is augmented by 7 more. The more entities in a sentence were automatically annotated, the higher the expected intra-sentence recall. Not augmenting sentences with only one entity found at all and focusing on sentences with more than one entity, possibly even more than two entities, is a promising approach to increase intra-entity recall. However, this approach has no

effect on intra-entity recall.

Despite these problems and misleading assumptions, annotation of natural language financial reports via distant supervision using the associated XBRL file is possible at least to the demonstrated extent and XBRL distant supervision is a relatively new and specialized area of research where this work can provide a helpful starting point. Finally, another problem with financial statement annotation using distant supervision is the insufficient availability of data. Companies based in Germany are required to submit an XBRL structure to the Federal Gazette, but access to it, as described in the background chapter of this work, still leaves much room for improvement. On the one hand, it is understandable that the data is only released against payment, because a lot of human preparation is behind it. On the other hand, transparent and structured access to corporate data is important for a functioning economy. Companies could be forced to publish XBRL data in a transparent and uniform manner. Otherwise, in addition to the time-consuming pre-processing, bureaucratic hurdles arise during processing.

8. Conclusion

This thesis aimed to investigate whether interrelated financial entities and values can be extracted from natural language text and whether the model for that can be trained using a dataset which was generated algorithmically with distant supervision instead of human annotated data.

Using the extended joint entity detection and relation extraction model, financial entities and values are correctly extracted as pairs with an F1 score of 82 from the test dataset when trained with a manually annotated and augmented dataset. This is the final quantitative empirical answer to the research question of whether financial entities and values can be extracted from natural text. Automated annotation of training data using XBRL distant supervision accordingly works with an F0.25 score of 90, which is the answer to the research question of whether the training dataset can be created by machine using distant supervision instead of human annotated data. However, the answer must be constrained in that the joint entity detection and relation extraction model produces absolutely about 14 percent better results with the manually annotated dataset. The distant supervision dataset therefore is not used to train the final model which was the initial goal of this work. The annotation and augmentation of a small dataset has been shown to be more effective in the experiments conducted. Data augmentation and the use of a refined BERT language model for the application domain each proved helpful to the final result with both datasets.

The evaluation of the presented joint entity detection and relation extraction model indicate that encoding explicit linguistic information into the model in addition to the contextualized word embeddings of the language model provide a slight improvement. The absolute improvement is about 2 percent on a standard dataset. The extent of the improvement indicates that the word embeddings already implicitly have extensive linguistic information and the explicit encoding is to be classified merely as an additional aid. Nevertheless, two percent is a step in the right direction and the published code can be adapted to add more features to the model.

The research contributions besides the results of the experiments are also the code of the extended joint entity detection and relation extraction model, the method and corresponding code for annotating XBRL entities in text using distant supervision, the XBRL data augmentation method and script, the manually annotated datasets with financial

entities and values, and the refined BERT model trained on financial reports. These artifacts are intended to make the experiments traceable, reproducible and adaptable.

For the application purpose in the context of the annual financial statement audit, the model is a helpful artifact for support. The dataset created with distant supervision still offers potential for improvement and further approaches can be investigated and compared. However, since this use case of XBRL information retrieval for distant supervision is quite new in research, the results are initially acceptable and a basis for further exploration. Furthermore, the joint entity detection model can be enriched with additional information such as the distance between two entities and the extraction use case can be extended in the long run to other entities like the previous years value of an entity in order to further increase the value of the model as a practical artifact.

Bibliography

- Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Literature review for Language and Statistics II* 2:1–15. (Cited on page 14).
- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2895–2905. Florence, Italy. (Cited on page 21).
- Eduardo Brito, Rafet Sifa, Christian Bauckhage, Rüdiger Loitz, Uwe Lohmeier, and Christin Pünt. 2019. A hybrid AI tool to extract key performance indicators from financial reports for benchmarking. In *Proceedings of the ACM Symposium on Document Engineering 2019*, 1–4. Berlin, Germany. (Cited on page 22).
- Bundesanzeiger Verlag GmbH. 2022a. B2B company data from primary sources. Accessed January 19, 2022. <https://www.validatis.de/en/solutions/b2b-data/>. (Cited on page 10).
- Bundesanzeiger Verlag GmbH. 2022b. Bundesanzeiger. Accessed January 19, 2022. <https://www.bundesanzeiger.de/pub/de/start?0>. (Cited on page 9).
- Bundesanzeiger Verlag GmbH. 2022c. XML-Schemata für Jahresabschlüsse. Accessed January 8, 2022. https://publikations-plattform.de/sp/service?page.navid=to_tech_std_annual_xml_scheme. (Cited on pages 38 sq.).
- Bundesministerium der Finanzen. 2018. Das Projekt E-Bilanz - ein wichtiger Baustein der Digitalisierung des Besteuerungsverfahrens. Accessed January 4, 2022. <https://www.bundesfinanzministerium.de/Monatsberichte/2018/08/Inhalte/Kapitel-3-Analysen/3-4-Das-Projekt-E-Bilanz.html>. (Cited on pages 6, 40).
- Bundesministerium für Finanzen. 2021. Veröffentlichung der Taxonomien 6.5 vom 14. April 2021. Accessed December 30, 2021. https://www.bundesfinanzministerium.de/Content/DE/Downloads/BMF_Schreiben/Steuerarten/Einkommensteuer/2021-07-09-e-bilanz-veroeffentlichung-der-taxonomien-6-5-vom-14-april-2021.pdf?__blob=publicationFile&v=1. (Cited on pages 6, 8, 37).
- Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, 724–731. Vancouver, British Columbia, Canada. (Cited on pages 30, 69).
- John Carroll. 2003. Parsing. Chap. 12 in *The Oxford Handbook of Computational Linguistics*, edited by Ruslan Mitkov, 233–248. (Cited on page 13).

-
- Clayton Chapman, Lars Hillebrand, Marc Robin Stenzel, Tobias Deusser, Christian Bauckhage, and Rafet Sifa. 2021. Towards Generating Financial Reports From Table Data Using Transformers, (cited on page 23).
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Doha, Qatar. (Cited on page 15).
- Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 423–429. Barcelona, Spain. (Cited on page 29).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, 4171–4186. Minneapolis, Minnesota. (Cited on pages 15 sq.).
- Kalpiti Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5308–5314. Florence, Italy. (Cited on page 20).
- AnHai Doan, Alon Halevy, and Zachary Ives. 2012. Principles of Data Integration. 1st. San Francisco, CA, USA. (Cited on pages 10 sq.).
- Markus Eberts and Adrian Ulges. 2019. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. In *Proceedings of the 24th European Conference on Artificial Intelligence*, 2006–2013. Online and Santiago de Compostela, Spain. (Cited on pages 21, 25 sq., 31).
- Markus Eberts and Adrian Ulges. 2021. An End-to-end Model for Entity-level Relation Extraction using Multi-instance Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 3650–3660. (Cited on page 19).
- Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 968–988. Online. (Cited on page 61).
- Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. RelEx—Relation extraction using dependency parse trees. *Bioinformatics* 23 (3): 365–371. (Cited on page 29).
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan. (Cited on pages 13, 52).
- Shirley Gregor and Alan R. Hevner. 2013. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly* 37 (2): 337–355. (Cited on page 3).
-

-
- Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2537–2547. Osaka, Japan. (Cited on pages 19 sq.).
- Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics* 45 (5): 885–892. (Cited on page 21).
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8342–8360. Online. (Cited on page 66).
- Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design Science in Information Systems Research. *MIS Quarterly* 28 (1): 75–105. (Cited on page 3).
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* (Cambridge, MA, USA) 9, no. 8 (November): 1735–1780. (Cited on page 15).
- Yi Yao Huang and William Yang Wang. 2017. Deep Residual Learning for Weakly-Supervised Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1803–1807. Copenhagen, Denmark. (Cited on page 14).
- Patrick Kahardipraja, Olena Vyshnevskya, and Sharid Loáiciga. 2020. Exploring span representations in neural coreference resolution. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, 32–41. Online. (Cited on page 26).
- Siti Sakira Kamaruddin, Abdul Razak Hamdan, Azuraliza Abu Bakar, and Fauzias Mat Nor. 2009. Automatic extraction of performance indicators from financial statements. In *2009 International Conference on Electrical Engineering and Informatics*, 2:348–350. Bangi, Malaysia: IEEE. (Cited on page 21).
- Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1411–1420. CIKM ’15. Melbourne, Australia. (Cited on page 12).
- Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 188–197. Copenhagen, Denmark. (Cited on pages 26, 69).
- Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1340–1350. Florence, Italy. (Cited on page 20).
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3219–3232. Brussels, Belgium. (Cited on page 21).
-

- Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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)*, 3036–3046. Minneapolis, Minnesota. (Cited on page 21).
- Youmi Ma, Tatsuya Hiraoka, and Naoaki Okazaki. 2022. Named entity recognition and relation extraction using enhanced table filling by Contextualized Representations. *Journal of Natural Language Processing* 29 (1): 187–223. (Cited on page 20).
- Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. 2008. Named entity recognition approaches. *International Journal of Computer Science and Network Security* 8 (2): 339–344. (Cited on page 14).
- Salvatore T. March and Gerald F. Smith. 1995. Design and natural science research on information technology. *Decision Support Systems* 15 (4): 251–266. (Cited on page 3).
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. Lake Tahoe, Nevada. (Cited on page 12).
- Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*. Scottsdale, Arizona, USA. (Cited on page 12).
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 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*, 1003–1011. Suntec, Singapore. (Cited on page 15).
- Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)* 54 (1): 1–39. (Cited on page 19).
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543. Doha, Qatar. (Cited on page 13).
- Karl Popper. 2005. The logic of scientific discovery. (Cited on page 4).
- S. Ravichandiran. 2021. Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT. (Cited on page 16).
- Rechenzentrum der Finanzverwaltung des Landes Nordrhein-Westfalen (Körperschaft des öffentlichen Rechts). 2021. Schnittstellen zur E-Bilanz § 5b EStG. Accessed December 31, 2021. <http://esteuer.de/#finanzantrag>. (Cited on pages 8, 40).
- Christian Riege, Jan Saat, and Tobias Bucher. 2009. Systematisierung von Evaluationsmethoden in der gestaltungsorientierten Wirtschaftsinformatik. In *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik*, 69–86. (Cited on page 4).

- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics* 8:842–866. (Cited on pages 16 sq.).
- Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, 1–8. Boston, Massachusetts, USA. (Cited on page 31).
- Alexander Rush. 2018. The Annotated Transformer. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 52–60. Melbourne, Australia. (Cited on pages 16 sq.).
- TYSS Santosh, Prantika Chakraborty, Sudakshina Dutta, Debarshi Kumar Sanyal, and Partha Pratim Das. 2021. Joint Entity and Relation Extraction from Scientific Documents: Role of Linguistic Information and Entity Types. In *Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) co-located with JCDL 2021*, 3004:15–19. Online. (Cited on pages 21, 69).
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725. Berlin, Germany. (Cited on page 25).
- Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404:132306. (Cited on page 15).
- Rafet Sifa, Anna Ladi, Maren Pielka, Rajkumar Ramamurthy, Lars Hillebrand, Birgit Kirsch, David Biesner, Robin Stenzel, Thiago Bell, Max Lübbering, et al. 2019. Towards automated auditing with machine learning. In *Proceedings of the ACM Symposium on Document Engineering 2019*, 1–4. Berlin, Germany. (Cited on page 22).
- Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* 24 (4): 35–43. (Cited on page 41).
- Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. 2006. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In *AI 2006: Advances in Artificial Intelligence*, edited by Abdul Sattar and Byeong-ho Kang, 1015–1021. Berlin, Heidelberg. (Cited on page 41).
- Stanford NLP Group. 2022. Stanford Parser. Accessed June 15, 2022. <https://nlp.stanford.edu/software/lex-parser.shtml>. (Cited on page 13).
- Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let’s Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3689–3701. Online. (Cited on page 19).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30. (Cited on page 15).
-

-
- John Venable, Jan Pries-Heje, and Richard Baskerville. 2012. A comprehensive framework for evaluation in design science research. In *International conference on design science research in information systems*, 423–438. Las Vegas, NV, USA: Springer. (Cited on page 4).
- Atro Voutilainen. 2003. Part-of-Speech Tagging. Chap. 11 in *The Oxford Handbook of Computational Linguistics*, edited by Ruslan Mitkov, 219–232. (Cited on page 13).
- David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 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)*, 5784–5789. Hong Kong, China. (Cited on pages 21, 26).
- Jue Wang and Wei Lu. 2020. Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1706–1721. Online. (Cited on pages 20, 69).
- Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UniRE: A Unified Label Space for Entity Relation Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 220–231. Online. (Cited on page 20).
- Thomas Wilde and Thomas Hess. 2006. Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung. Technical report. (Cited on page 4).
- XBRL Deutschland e. V. 2021a. Was ist XBRL? Accessed December 20, 2021. <https://de.xbrl.org/was-ist-xbrl/>. (Cited on page 6).
- XBRL Deutschland e. V. 2021b. Was ist XBRL? - Blick unter die Motorhaube. Accessed December 29, 2021. <https://de.xbrl.org/was-ist-xbrl/blick-unter-die-motorhaube/>. (Cited on pages 7, 34).
- XBRL International Inc. 2021a. Taxonomies. Accessed December 30, 2021. <https://www.xbrl.org/the-standard/what/taxonomies/>. (Cited on page 8).
- XBRL International Inc. 2021b. XBRL Taxonomy Registry. Accessed December 30, 2021. <https://taxonomies.xbrl.org/>. (Cited on page 8).
- Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed Levitated Marker for Entity and Relation Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4904–4917. Dublin, Ireland. (Cited on page 21).
- Zexuan Zhong and Danqi Chen. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 50–61. Online. (Cited on pages 19 sq.).
-

Affidavit

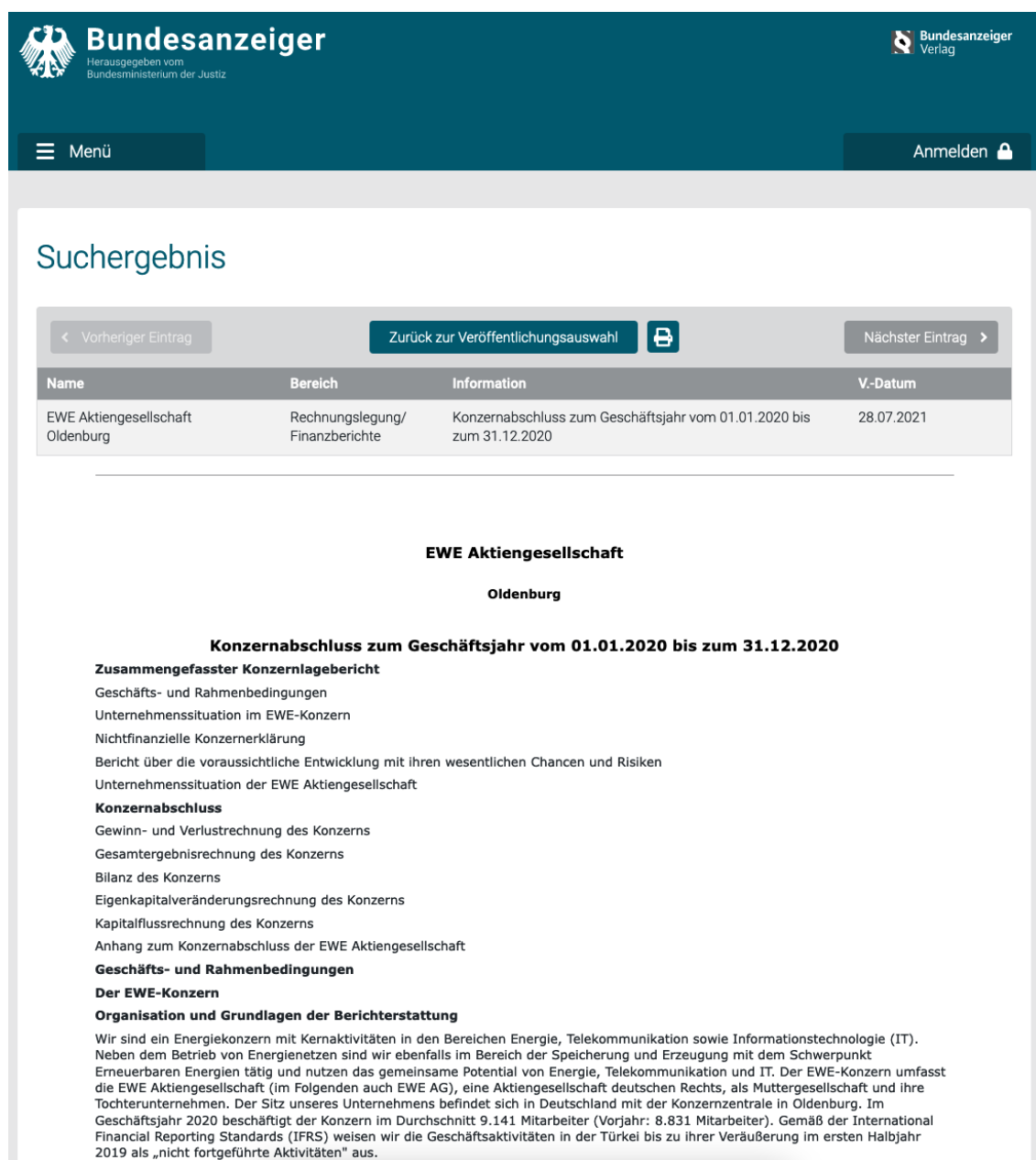
Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudien-
gang Wirtschaftsinformatik selbstständig verfasst und keine anderen als die angegebe-
nen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-
Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichun-
gen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin,
dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe
und die eingereichte schriftliche Fassung der elektronischen Abgabe entspricht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einver-
standen.

Hamburg, den 29.06.2022 Unterschrift: F. Rausch

A. Federal gazette publications

A.1. Specific presentation from the Federal Gazette example



Bundesanzeiger
Herausgegeben vom
Bundesministerium der Justiz

Bundesanzeiger
Verlag

Menü Anmelden

Suchergebnis

< Vorheriger Eintrag Zurück zur Veröffentlichungsauswahl Nächster Eintrag >

Name	Bereich	Information	V.-Datum
EWE Aktiengesellschaft Oldenburg	Rechnungslegung/ Finanzberichte	Konzernabschluss zum Geschäftsjahr vom 01.01.2020 bis zum 31.12.2020	28.07.2021

EWE Aktiengesellschaft
Oldenburg

Konzernabschluss zum Geschäftsjahr vom 01.01.2020 bis zum 31.12.2020

Zusammengefasster Konzernlagebericht
Geschäfts- und Rahmenbedingungen
Unternehmenssituation im EWE-Konzern
Nichtfinanzielle Konzernklärung
Bericht über die voraussichtliche Entwicklung mit ihren wesentlichen Chancen und Risiken
Unternehmenssituation der EWE Aktiengesellschaft

Konzernabschluss
Gewinn- und Verlustrechnung des Konzerns
Gesamtergebnisrechnung des Konzerns
Bilanz des Konzerns
Eigenkapitalveränderungsrechnung des Konzerns
Kapitalflussrechnung des Konzerns
Anhang zum Konzernabschluss der EWE Aktiengesellschaft

Geschäfts- und Rahmenbedingungen
Der EWE-Konzern
Organisation und Grundlagen der Berichterstattung
Wir sind ein Energiekonzern mit Kernaktivitäten in den Bereichen Energie, Telekommunikation sowie Informationstechnologie (IT). Neben dem Betrieb von Energienetzen sind wir ebenfalls im Bereich der Speicherung und Erzeugung mit dem Schwerpunkt Erneuerbaren Energien tätig und nutzen das gemeinsame Potential von Energie, Telekommunikation und IT. Der EWE-Konzern umfasst die EWE Aktiengesellschaft (im Folgenden auch EWE AG), eine Aktiengesellschaft deutschen Rechts, als Muttergesellschaft und ihre Tochterunternehmen. Der Sitz unseres Unternehmens befindet sich in Deutschland mit der Konzernzentrale in Oldenburg. Im Geschäftsjahr 2020 beschäftigt der Konzern im Durchschnitt 9.141 Mitarbeiter (Vorjahr: 8.831 Mitarbeiter). Gemäß der International Financial Reporting Standards (IFRS) weisen wir die Geschäftsaktivitäten in der Türkei bis zu ihrer Veräußerung im ersten Halbjahr 2019 als „nicht fortgeführte Aktivitäten“ aus.

Figure A.1.: Specific presentation from the Federal Gazette for the company EWE AG as an example

A.2. Official publication file example

Geschäfts- und Rahmenbedingungen

Der EWE-Konzern

Organisation und Grundlagen der Berichterstattung

Wir sind ein Energiekonzern mit Kernaktivitäten in den Bereichen Energie, Telekommunikation sowie Informationstechnologie (IT). Die Konzernzentrale ist in Oldenburg. Im Geschäftsjahr 2020 beschäftigt der Konzern im Durchschnitt 9.141 Mitarbeiter (Vorjahr: 8.831).

Beschreibung der Geschäftstätigkeit

Segment Erneuerbare Energien

Im Bereich Erneuerbare planen, bauen und betreiben wir Windenergieanlagen zur regenerativen Energieerzeugung, teilweise im R

Segment Infrastruktur

Im Bereich Netze betreiben wir Strom- und Erdgasnetze im Ems-Weser-Elbe-Gebiet sowie Erdgasnetze in Brandenburg, auf Rügen und im Telekommunikationsnetz von 54,2 Tsd. km (Vorjahr: 51,9 Tsd. km). Der Breitbandausbau in der ländlich geprägten Region im F

Im Bereich Gasspeicher errichten, erwerben und betreiben wir Anlagen zur Lagerung sowie zur Ein- und Ausspeicherung von gas

Segment Markt

Der Bereich Energie und Telekommunikation kombiniert den Vertrieb von Energie- und Telekommunikationsprodukten. Der Fokus liegt auf innovativen Lösungen, neue Geschäftsmöglichkeiten eröffnen.

Der Bereich Handel bündelt Dienstleistungen im Rahmen der Beschaffung und Vermarktung von Strom und Gas. Darüber hinaus

Figure A.2.: Official presentation file (XHTML) from the company EWE AG as an example

A.3. Official publication directory overview example

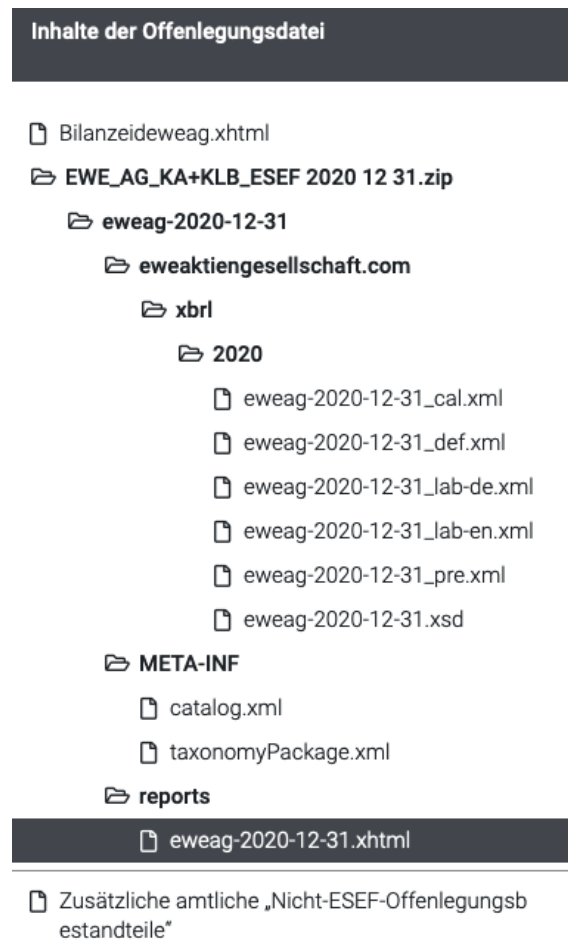


Figure A.3.: Official presentation directory structure from the company EWE AG as an example

B. Raw data basis examples

B.1. XBRL data

Listing B.1: Example XBRL file

```

1 <?xml version="1.0" encoding="UTF-8"?><!-- Data created by
   Bundesanzeiger Verlag / BDS --><!-- Trust Rank: green#green#green
   -->
2 <xbrl xmlns="http://www.xbrl.org/2003/instance" xmlns:link="http://www.
   xbrl.org/2003/linkbase" xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:de-gaap
   -ci="http://www.xbrl.de/taxonomies/de-gaap-ci-2016-04-01" xmlns:de-
   gcd="http://www.xbrl.de/taxonomies/de-gcd-2016-04-01">
3 <link:schemaRef xlink:type="simple" xlink:arcrole="http://www.w3.org
   /1999/xlink/properties/linkbase" xlink:href="http://www.xbrl.de/
   taxonomies/de-gaap-ci-2016-04-01/de-gaap-ci-2016-04-01-shell.xsd"/>
4 <link:schemaRef xlink:type="simple" xlink:arcrole="http://www.w3.org
   /1999/xlink/properties/linkbase" xlink:href="http://www.xbrl.de/
   taxonomies/de-gcd-2016-04-01/de-gcd-2016-04-01-shell.xsd"/>
5
6
7
8 <unit id="id210512038457_UNIT"><measure xmlns:iso4217="http://www.xbrl.
   org/2003/iso4217">iso4217:EUR</measure></unit>
9
10 <!-- ##### CONTEXT ##### -->
11 <context id="id210512038457_CY_INSTANT">
12 <entity><identifier scheme="http://www.bundesanzeiger.de/ebanz">
   210512038457</identifier>
13 </entity><period>
14 <instant>2020-12-31</instant>
15 </period></context>
16 <context id="id210512038457_CY_DURATION">
17 <entity><identifier scheme="http://www.bundesanzeiger.de/ebanz">
   210512038457</identifier>
18 </entity><period>
19 <startDate>2020-01-01</startDate>
20 <endDate>2020-12-31</endDate>
21 </period></context>

```

```

22 <context id="id210512038457_PY_INSTANT">
23 <entity><identifizier scheme="http://www.bundesanzeiger.de/ebanz">
    210512038457</identifizier>
24 </entity><period>
25 <instant>2019-12-31</instant>
26 </period></context>
27 <context id="id210512038457_PY_DURATION">
28 <entity><identifizier scheme="http://www.bundesanzeiger.de/ebanz">
    210512038457</identifizier>
29 </entity><period>
30 <startDate>2019-01-01</startDate>
31 <endDate>2019-12-31</endDate>
32 </period></context>
33
34 <de-gcd:genInfo.doc.id.generationDate contextRef="
    id210512038457_CY_DURATION">2021-06-10</de-gcd:genInfo.doc.id.
    generationDate>
35 <de-gcd:genInfo.company.id.idNo>
36 <de-gcd:genInfo.company.id.idNo.type.companyId.EN contextRef="
    id210512038457_CY_DURATION">100247905</de-gcd:genInfo.company.id.
    idNo.type.companyId.EN>
37 </de-gcd:genInfo.company.id.idNo>
38 <de-gcd:genInfo.company.id.sizeClass>
39 <de-gcd:genInfo.company.id.sizeClass.sizeClass.KK contextRef="
    id210512038457_CY_DURATION"/>
40 </de-gcd:genInfo.company.id.sizeClass>
41 <de-gcd:genInfo.doc.rev.versionNo contextRef="
    id210512038457_CY_DURATION">210512038457</de-gcd:genInfo.doc.rev.
    versionNo>
42 <de-gcd:genInfo.report.id.reportStatus.reportStatus.dateofdisclosure
    contextRef="id210512038457_CY_DURATION">2021-06-09</de-gcd:genInfo.
    report.id.reportStatus.reportStatus.dateofdisclosure>
43 <de-gcd:genInfo.report.period.reportPeriodBegin contextRef="
    id210512038457_CY_DURATION">2020-01-01</de-gcd:genInfo.report.
    period.reportPeriodBegin>
44 <de-gcd:genInfo.report.period.reportPeriodEnd contextRef="
    id210512038457_CY_DURATION">2020-12-31</de-gcd:genInfo.report.
    period.reportPeriodEnd>
45 <de-gcd:genInfo.report.id.reportType>
46 <de-gcd:genInfo.report.id.reportType.reportType.JA contextRef="
    id210512038457_CY_DURATION"/>
47 </de-gcd:genInfo.report.id.reportType>
48 <de-gcd:genInfo.company.id.name contextRef="id210512038457_CY_DURATION"
    >VTG Aktiengesellschaft</de-gcd:genInfo.company.id.name>
49 <de-gcd:genInfo.report.id.consolidationRange>

```

```

50     <de-gcd:genInfo.report.id consolidationRange consolidationRange.EA
      contextRef="id210512038457_CY_DURATION"/>
51 </de-gcd:genInfo.report.id consolidationRange>
52 <de-gcd:genInfo.report.id revisionStatus>
53     <de-gcd:genInfo.report.id revisionStatus.revisionStatus.E
      contextRef="id210512038457_CY_DURATION"/>
54 </de-gcd:genInfo.report.id revisionStatus>
55 <de-gcd:genInfo.company.id.location contextRef="
      id210512038457_CY_DURATION">Hamburg</de-gcd:genInfo.company.id.
      location>
56 <de-gcd:genInfo.company.id.location.street contextRef="
      id210512038457_CY_DURATION">Nagelsweg 34</de-gcd:genInfo.company.id
      .location.street>
57 <de-gcd:genInfo.company.id.location.zipCode contextRef="
      id210512038457_CY_DURATION">20097</de-gcd:genInfo.company.id.
      location.zipCode>
58 <de-gcd:genInfo.company.id.location.city contextRef="
      id210512038457_CY_DURATION">Hamburg</de-gcd:genInfo.company.id.
      location.city>
59 <de-gcd:genInfo.company.id.location.country contextRef="
      id210512038457_CY_DURATION">Deutschland</de-gcd:genInfo.company.id.
      location.country>
60 <de-gcd:genInfo.company.id.industry>
61     <de-gcd:genInfo.company.id.industry.keyType>
62         <de-gcd:genInfo.company.id.industry.keyType.industryKey.WZ2008
          contextRef="id210512038457_CY_DURATION"/>
63     </de-gcd:genInfo.company.id.industry.keyType>
64     <de-gcd:genInfo.company.id.industry.keyEntry contextRef="
      id210512038457_CY_DURATION">70101</de-gcd:genInfo.company.id.
      industry.keyEntry>
65 </de-gcd:genInfo.company.id.industry>
66 <de-gcd:genInfo.company.id.Incorporation.court contextRef="
      id210512038457_CY_DURATION">Hamburg</de-gcd:genInfo.company.id.
      Incorporation.court>
67 <de-gcd:genInfo.company.id.Incorporation.Type>
68     <de-gcd:genInfo.company.id.Incorporation.Type.Type.HR contextRef="
      id210512038457_CY_DURATION"/>
69 </de-gcd:genInfo.company.id.Incorporation.Type>
70 <de-gcd:genInfo.company.id.Incorporation.number contextRef="
      id210512038457_CY_DURATION">98591</de-gcd:genInfo.company.id.
      Incorporation.number>
71
72 <!-- #### data-section #### -->
73 <de-gaap-ci:bs.ass contextRef="id210512038457_CY_INSTANT" decimals="2"
      unitRef="id210512038457_UNIT">1066624466.94</de-gaap-ci:bs.ass>

```

```

74 <de-gaap-ci:bs.ass.currAss contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">373668620.69</de-gaap-
    ci:bs.ass.currAss>
75 <de-gaap-ci:bs.ass.currAss.cashEquiv contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">20987.70</de-gaap-ci:bs.ass.currAss.cashEquiv>
76 <de-gaap-ci:bs.ass.currAss.cashEquiv.bank contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">20987.70</de-gaap-ci:bs.ass.currAss.cashEquiv.
    bank>
77 <de-gaap-ci:bs.ass.currAss.receiv contextRef="id210512038457_CY_INSTANT
    " decimals="2" unitRef="id210512038457_UNIT">373647632.99</de-gaap-
    ci:bs.ass.currAss.receiv>
78 <de-gaap-ci:bs.ass.currAss.receiv.affil contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">372709351.10</de-gaap-ci:bs.ass.currAss.receiv
    .affil>
79 <de-gaap-ci:bs.ass.currAss.receiv.other contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">938281.89</de-gaap-ci:bs.ass.currAss.receiv.
    other>
80 <de-gaap-ci:bs.ass.currAss.receiv.trade contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">0.00</de-gaap-ci:bs.ass.currAss.receiv.trade>
81 <de-gaap-ci:bs.ass.fixAss contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">692018286.80</de-gaap-
    ci:bs.ass.fixAss>
82 <de-gaap-ci:bs.ass.fixAss.fin contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">692018286.80</de-gaap-
    ci:bs.ass.fixAss.fin>
83 <de-gaap-ci:bs.ass.fixAss.fin.loansToAffil contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">6097106.89</de-gaap-ci:bs.ass.fixAss.fin.
    loansToAffil>
84 <de-gaap-ci:bs.ass.fixAss.fin.otherLoans contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">0.00</de-gaap-ci:bs.ass.fixAss.fin.otherLoans>
85 <de-gaap-ci:bs.ass.fixAss.fin.particip contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">15000000.00</de-gaap-ci:bs.ass.fixAss.fin.
    particip>
86 <de-gaap-ci:bs.ass.fixAss.fin.sharesInAffil contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">670921179.91</de-gaap-ci:bs.ass.fixAss.fin.
    sharesInAffil>

```

```
87 <de-gaap-ci:bs.ass.prepaidExp contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">937559.45</de-gaap-ci:bs
    .ass.prepaidExp>
88 <de-gaap-ci:bs.eqLiab contextRef="id210512038457_CY_INSTANT" decimals="
    2" unitRef="id210512038457_UNIT">1066624466.94</de-gaap-ci:bs.
    eqLiab>
89 <de-gaap-ci:bs.eqLiab.accruals contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">45972596.70</de-gaap-
    ci:bs.eqLiab.accruals>
90 <de-gaap-ci:bs.eqLiab.accruals.other contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">6020177.81</de-gaap-ci:bs.eqLiab.accruals.
    other>
91 <de-gaap-ci:bs.eqLiab.accruals.pensions contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">14460428.00</de-gaap-ci:bs.eqLiab.accruals.
    pensions>
92 <de-gaap-ci:bs.eqLiab.accruals.tax contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">25491990.89</de-gaap-ci:bs.eqLiab.accruals.tax
    >
93 <de-gaap-ci:bs.eqLiab.equity contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">711940703.16</de-gaap-
    ci:bs.eqLiab.equity>
94 <de-gaap-ci:bs.eqLiab.equity.capRes contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">615516813.65</de-gaap-ci:bs.eqLiab.equity.
    capRes>
95 <de-gaap-ci:bs.eqLiab.equity.profitLoss contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">62190298.51</de-gaap-ci:bs.eqLiab.equity.
    profitLoss>
96 <de-gaap-ci:bs.eqLiab.equity.subscribed contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">34233591.00</de-gaap-ci:bs.eqLiab.equity.
    subscribed>
97 <de-gaap-ci:bs.eqLiab.equity.subscribed.corp contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">34233591.00</de-gaap-ci:bs.eqLiab.equity.
    subscribed.corp>
98 <de-gaap-ci:bs.eqLiab.liab contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">308711167.08</de-gaap-
    ci:bs.eqLiab.liab>
99 <de-gaap-ci:bs.eqLiab.liab.assocComp contextRef="
    id210512038457_CY_INSTANT" decimals="2" unitRef="
    id210512038457_UNIT">307022766.09</de-gaap-ci:bs.eqLiab.liab.
```

```

    assocComp>
100 <de-gaap-ci:bs.eqLiab.liab.other contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">886626.19</de-gaap-
    ci:bs.eqLiab.liab.other>
101 <de-gaap-ci:bs.eqLiab.liab.trade contextRef="id210512038457_CY_INSTANT"
    decimals="2" unitRef="id210512038457_UNIT">801774.80</de-gaap-
    ci:bs.eqLiab.liab.trade>
102 <de-gaap-ci:incomeUse.gainLoss contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">62190298.51</de-gaap-
    ci:incomeUse.gainLoss>
103 <de-gaap-ci:incomeUse.gainLoss.netIncome contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">61054487.94</de-gaap-ci:incomeUse.gainLoss.
    netIncome>
104 <de-gaap-ci:incomeUse.gainLoss.retainedEarningsPrevYear contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">1135810.57</de-gaap-ci:incomeUse.gainLoss.
    retainedEarningsPrevYear>
105 <de-gaap-ci:is.netIncome contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">61054487.94</de-gaap-
    ci:is.netIncome>
106 <de-gaap-ci:is.netIncome.eat contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">60967900.50</de-gaap-
    ci:is.netIncome.eat>
107 <de-gaap-ci:is.netIncome.otherTaxes contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">-86587.44</de-gaap-ci:is.netIncome.otherTaxes>
108 <de-gaap-ci:is.netIncome.regular.fin contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">75400020.43</de-gaap-ci:is.netIncome.regular.
    fin>
109 <de-gaap-ci:is.netIncome.regular.fin.netInterest.expenses contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">10642941.39</de-gaap-ci:is.netIncome.regular.
    fin.netInterest.expenses>
110 <de-gaap-ci:is.netIncome.regular.fin.netInterest.income contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">8854235.18</de-gaap-ci:is.netIncome.regular.
    fin.netInterest.income>
111 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.amortFinanc
    contextRef="id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">11200000.00</de-gaap-ci:is.netIncome.regular.
    fin.netParticipation.amortFinanc>
112 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.amortFinanc.
    financials contextRef="id210512038457_CY_DURATION" decimals="2"
    unitRef="id210512038457_UNIT">11200000.00</de-gaap-ci:is.netIncome.

```

```

regular.fin.netParticipation.amortFinanc.financials>
113 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.
    earningProfSharing contextRef="id210512038457_CY_DURATION" decimals
    ="2" unitRef="id210512038457_UNIT">88648497.14</de-gaap-ci:is.
    netIncome.regular.fin.netParticipation.earningProfSharing>
114 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.
    earningProfSharing.other contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">88648497.14</de-gaap-
    ci:is.netIncome.regular.fin.netParticipation.earningProfSharing.
    other>
115 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.earnings
    contextRef="id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">7248615.91</de-gaap-ci:is.netIncome.regular.
    fin.netParticipation.earnings>
116 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.earningSecurities
    contextRef="id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">0.00</de-gaap-ci:is.netIncome.regular.fin.
    netParticipation.earningSecurities>
117 <de-gaap-ci:is.netIncome.regular.fin.netParticipation.loss contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">7508386.41</de-gaap-ci:is.netIncome.regular.
    fin.netParticipation.loss>
118 <de-gaap-ci:is.netIncome.regular.operatingTC contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">-8372757.93</de-gaap-ci:is.netIncome.regular.
    operatingTC>
119 <de-gaap-ci:is.netIncome.regular.operatingTC.grossTradingProfit
    contextRef="id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">27205047.15</de-gaap-ci:is.netIncome.regular.
    operatingTC.grossTradingProfit>
120 <de-gaap-ci:is.netIncome.regular.operatingTC.grossTradingProfit.
    materialServices contextRef="id210512038457_CY_DURATION" decimals="
    2" unitRef="id210512038457_UNIT">1788247.93</de-gaap-ci:is.
    netIncome.regular.operatingTC.grossTradingProfit.materialServices>
121 <de-gaap-ci:is.netIncome.regular.operatingTC.grossTradingProfit.
    materialServices.services contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">1788247.93</de-gaap-
    ci:is.netIncome.regular.operatingTC.grossTradingProfit.
    materialServices.services>
122 <de-gaap-ci:is.netIncome.regular.operatingTC.grossTradingProfit.
    totalOutput contextRef="id210512038457_CY_DURATION" decimals="2"
    unitRef="id210512038457_UNIT">10415994.36</de-gaap-ci:is.netIncome.
    regular.operatingTC.grossTradingProfit.totalOutput>
123 <de-gaap-ci:is.netIncome.regular.operatingTC.grossTradingProfit.
    totalOutput.netSales contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">10415994.36</de-gaap-

```

```

    ci:is.netIncome.regular.operatingTC.grossTradingProfit.totalOutput.
    netSales>
124 <de-gaap-ci:is.netIncome.regular.operatingTC.otherCost contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">24229031.64</de-gaap-ci:is.netIncome.regular.
    operatingTC.otherCost>
125 <de-gaap-ci:is.netIncome.regular.operatingTC.otherOpRevenue contextRef="
    "id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">18577300.72</de-gaap-ci:is.netIncome.regular.
    operatingTC.otherOpRevenue>
126 <de-gaap-ci:is.netIncome.regular.operatingTC.staff contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">11348773.44</de-gaap-ci:is.netIncome.regular.
    operatingTC.staff>
127 <de-gaap-ci:is.netIncome.regular.operatingTC.staff.salaries contextRef="
    "id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">8174452.97</de-gaap-ci:is.netIncome.regular.
    operatingTC.staff.salaries>
128 <de-gaap-ci:is.netIncome.regular.operatingTC.staff.social contextRef="
    id210512038457_CY_DURATION" decimals="2" unitRef="
    id210512038457_UNIT">3174320.47</de-gaap-ci:is.netIncome.regular.
    operatingTC.staff.social>
129 <de-gaap-ci:is.netIncome.tax contextRef="id210512038457_CY_DURATION"
    decimals="2" unitRef="id210512038457_UNIT">6059362.00</de-gaap-
    ci:is.netIncome.tax>
130 </xbrl>

```

C. Dataset statistics and properties

C.1. XBRL entities

Table C.1.: XBRL entity references and frequency

de-gaap-ci:bs.ass	5604
de-gaap-ci:bs.ass.currass	5587
de-gaap-ci:bs.ass.currass.cashequiv	5289
de-gaap-ci:bs.ass.currass.inventory	2430
de-gaap-ci:bs.ass.currass.receive	5414
de-gaap-ci:bs.ass.currass.receive.above1year	271
de-gaap-ci:bs.eqliab	5604
de-gaap-ci:bs.eqliab.accruals	5493
de-gaap-ci:bs.eqliab.equity	5596
de-gaap-ci:bs.eqliab.equity.subscribed	5351
de-gaap-ci:bs.eqliab.equity.subscribed.limitedliablepartners	288
de-gaap-ci:bs.eqliab.liab	5454
de-gaap-ci:bs.eqliab.liab.upto1year	581
de-gaap-ci:bs.ass.fixass	5292
de-gaap-ci:bs.ass.fixass.fin	4753
de-gaap-ci:bs.ass.fixass.intan	3566
de-gaap-ci:bs.ass.fixass.tan	4159
de-gaap-ci:bs.ass.prepaidexp	4247
de-gaap-ci:bs.eqliab.equity.capres	3589
de-gaap-ci:bs.eqliab.equity.profitloss	2797
de-gaap-ci:bs.eqliab.equity.revenueres	2675
de-gaap-ci:bs.eqliab.equity.subscribed.corp	2351
de-gaap-ci:bs.eqliab.equity.subscribed.ownsharesdeducted	133
de-gaap-ci:bs.eqliab.pretaxres	279
de-gaap-ci:bs.eqliab.pretaxres.res	57
de-gaap-ci:bs.eqliab.pretaxres.res.subsidies	52
de-gaap-ci:incomeuse.gainloss	396
de-gaap-ci:incomeuse.gainloss.netincome	395

de-gaap-ci:incomeuse.gainloss.retainedearningsprevyear	253
de-gaap-ci:is.netincome	2288
de-gaap-ci:is.netincome.eat	386
de-gaap-ci:is.netincome.othertaxes	1594
de-gaap-ci:is.netincome.regular.fin	2258
de-gaap-ci:is.netincome.regular.operatingtc	2243
de-gaap-ci:is.netincome.regular.operatingtc.depramort	1888
de-gaap-ci:is.netincome.regular.operatingtc.depramort.fixass	1756
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit	2175
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices	1482
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput	1683
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange	822
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput.netsales	1679
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .ownwork	546
de-gaap-ci:is.netincome.regular.operatingtc.othercost	2245
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue	2064
de-gaap-ci:is.netincome.regular.operatingtc.staff	1954
de-gaap-ci:is.netincome.tax	1926
de-gaap-ci:bs.eqliab.equity.revenueres.legal	662
de-gaap-ci:bs.eqliab.equity.revenueres.other	946
de-gaap-ci:bs.ass.deficitnotcoveredbycapital	387
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.losslimitedliablepartners	10
de-gaap-ci:bs.eqliab.equity.netincome	2056
de-gaap-ci:bs.ass.currass.cashequiv.bank	828
de-gaap-ci:bs.ass.currass.receive.affil	2163
de-gaap-ci:bs.ass.currass.receive.other	2886
de-gaap-ci:bs.ass.deftax	460
de-gaap-ci:bs.ass.fixass.fin.loantoaffil	610
de-gaap-ci:bs.ass.fixass.fin.particip	1301
de-gaap-ci:bs.ass.fixass.fin.sharesinaffil	1907
de-gaap-ci:bs.ass.fixass.tan.landbuildings	1428
de-gaap-ci:bs.ass.fixass.tan.otherequipm	2032
de-gaap-ci:bs.eqliab.accruals.other	2924
de-gaap-ci:bs.eqliab.accruals.pensions	1535
de-gaap-ci:bs.eqliab.accruals.tax	2063
de-gaap-ci:bs.eqliab.liab.assoccomp	2070

de-gaap-ci:bs.eqliab.liab.bank	1755
de-gaap-ci:bs.eqliab.liab.other	2859
de-gaap-ci:bs.eqliab.liab.other.other	288
de-gaap-ci:bs.eqliab.liab.other.therofftax	1276
de-gaap-ci:bs.eqliab.liab.shareholders	444
de-gaap-ci:bs.eqliab.liab.trade	2678
de-gaap-ci:incomeuse.gainloss.additionrevenreserves	77
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses	2013
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses.assoc	325
de-gaap-ci:is.netincome.regular.fin.netinterest.income	2029
de-gaap-ci:is.netincome.regular.fin.netinterest.income.assoc	352
de-gaap-ci:is.netincome.regular.fin.netparticipation.earningprofsharing	542
de-gaap-ci:is.netincome.regular.fin.netparticipation.earningprofsharing.other	511
de-gaap-ci:is.netincome.regular.fin.netparticipation.earnings	1043
de-gaap-ci:is.netincome.regular.fin.netparticipation.earnings.groupcomp	194
de-gaap-ci:is.netincome.regular.fin.netparticipation.earningsecurities	640
de-gaap-ci:is.netincome.regular.fin.netparticipation.earningsecurities.assoc	68
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .services	979
de-gaap-ci:is.netincome.regular.operatingtc.staff.salaries	1625
de-gaap-ci:is.netincome.regular.operatingtc.staff.social	1611
de-gaap-ci:is.netincome.regular.operatingtc.staff.social.other	101
de-gaap-ci:is.netincome.regular.operatingtc.staff.social.pensions	397
de-gaap-ci:is.netincome.taxes	34
de-gaap-ci:bs.ass.currass.inventory.advpaympaid	424
de-gaap-ci:bs.ass.currass.inventory.finishedandmerch	1166
de-gaap-ci:bs.ass.currass.inventory.inprogress	828
de-gaap-ci:bs.ass.currass.inventory.inprogress.goods	386
de-gaap-ci:bs.ass.currass.inventory.material	1010
de-gaap-ci:bs.ass.currass.receiv.trade	1889
de-gaap-ci:bs.ass.currass.securities	1225
de-gaap-ci:bs.ass.fixass.fin.loanstparticip	150
de-gaap-ci:bs.ass.fixass.fin.otherloans	660
de-gaap-ci:bs.ass.fixass.fin.securities	481
de-gaap-ci:bs.ass.fixass.intan.concessionbrands	1728
de-gaap-ci:bs.ass.fixass.intan.goodwill	403
de-gaap-ci:bs.ass.fixass.tan.inconstradvpaym	1100
de-gaap-ci:bs.ass.fixass.tan.machinery	1020

de-gaap-ci:bs.eqliab.defincome	1476
de-gaap-ci:bs.eqliab.liab.advpaym	629
de-gaap-ci:bs.ass.currass.receive.particip	683
de-gaap-ci:bs.ass.currass.receive.shareholders	184
de-gaap-ci:bs.eqliab.liab.particip	559
de-gaap-ci:incomeuse.gainloss.dividendsplanned	59
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .material	1088
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .material.purchased	509
de-gaap-ci:bs.eqliab.equity.retainedearnings	1789
de-gaap-ci:bs.eqliab.liab.securities	270
de-gaap-ci:incomeuse.gainloss.releaseotherres	9
de-gaap-ci:incomeuse.gainloss.releaserevenreserves	18
de-gaap-ci:is.netincome.regular.fin.netparticipation.amortfinanc	836
de-gaap-ci:bs.ass.currass.inventory.inprogress.services	109
de-gaap-ci:bs.eqliab.equity.deficitnotcoveredbycapital	178
de-gaap-ci:incomeuse.gainloss.accumlossprevyear	117
de-gaap-ci:bs.eqliab.deftax	352
de-gaap-ci:is.netincome.regular.operatingtc.depramort.currass	122
de-gaap-ci:bs.eqliab.equity.reservespartnership	122
de-gaap-ci:bs.eqliab.otherspecres	318
de-gaap-ci:bs.eqliab.otherspecres.ownshares	52
de-gaap-ci:is.netincome.regular.fin.netparticipation.amortfinanc.financials	446
de-gaap-ci:is.netincome.regular.operatingtc.depramort.fixass.tan	96
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .material.rawmatconss	5
de-gaap-ci:bs.ass.currass.inventory.other	96
de-gaap-ci:bs.ass.surplusfromoffsetting	225
de-gaap-ci:is.netincome.regular	1902
de-gaap-ci:is.netincome.regular.fin.netparticipation	1224
de-gaap-ci:is.netincome.regular.fin.netinterest	1614
de-gaap-ci:is.netincome.extraord	529
de-gaap-ci:bs.ass.fixass.intan.advpaym	392
de-gaap-ci:bs.ass.fixass.fin.otherfinass	132
de-gaap-ci:bs.ass.fixass.fin.otherfinass.reinsurclaim	76
de-gaap-ci:bs.eqliab.equity.consolsurplus	26
de-gaap-ci:bs.eqliab.equity.currtransl	106

de-gaap-ci:bs.eqliab.otherspecres.subsidies	191
de-gaap-ci:is.netincome.extraord.income	196
de-gaap-ci:is.netincome.extraord.expenses	407
de-gaap-ci:bs.ass.fixass.intan.concessionbrands.software	234
de-gaap-ci:bs.ass.currass.inventory.advpaymreceived	182
de-gaap-ci:bs.ass.currass.securities.other	464
de-gaap-ci:bs.ass.fixass.tan.otherequipm.office	46
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.inprogres	106
de-gaap-ci:is.netincome.regular.operatingtc.depramort.fixass.intan	291
de-gaap-ci:is.netincome.incomesharing	281
de-gaap-ci:is.netincome.incomesharing.loss	61
de-gaap-ci:bs.ass.fixass.intan.concessionbrands.other	253
de-gaap-ci:bs.ass.fixass.intan.other	106
de-gaap-ci:bs.ass.fixass.tan.landbuildings.rigtequivalenttolandwithoutbuildings	188
de-gaap-ci:bs.ass.fixass.tan.machinery.technequipm	213
de-gaap-ci:bs.ass.currass.securities.ownshares	135
de-gaap-ci:bs.eqliab.equity.revenueres.forownshares	126
de-gaap-ci:bs.eqliab.liab.other.thereoffsocsec	851
de-gaap-ci:is.netincome.regular.fin.netparticipation.loss	366
de-gaap-ci:is.netincome.regular.fin.netparticipation.loss.other	162
de-gaap-ci:bs.eqliab.accruals.tax.deftax	4
de-gaap-ci:bs.ass.currass.inventory.finishedandmerch.goods	95
de-gaap-ci:bs.ass.fixass.intan.concessionbrands.trademarks	44
de-gaap-ci:bs.ass.prepaidexp.other	81
de-gaap-ci:is.netincome.regular.fin.netinterest.income.valuediscount	28
de-gaap-ci:is.netincome.regular.operatingtc.othercost.exchange	119
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue.exchange	104
de-gaap-ci:bs.eqliab.equity.subscribed.calledin	112
de-gaap-ci:is.netincome.incomesharing.gain	239
de-gaap-ci:is.netincome.incomesharing.gain.other	198
de-gaap-ci:bs.ass.currass.inventory.finishedandmerch.merchandise	335
de-gaap-ci:bs.eqliab.liab.advpaym.upto1year	69
de-gaap-ci:bs.eqliab.liab.assoccomp.upto1year	301
de-gaap-ci:bs.eqliab.liab.bank.upto1year	247
de-gaap-ci:bs.eqliab.liab.other.upto1year	504
de-gaap-ci:bs.eqliab.liab.trade.upto1year	464
de-gaap-ci:bs.eqliab.equity.deficitnotcovered	130

de-gaap-ci:bs.eqliab.otherspecres.paymforcapitalincrease	35
de-gaap-ci:bs.ass.fixass.tan.landbuildings.buildingsonnonownedland	113
de-gaap-ci:bs.ass.currass.receive.other.unpaidcapital	3
de-gaap-ci:bs.ass.currass.receive.other.above1year	261
de-gaap-ci:bs.eqliab.liab.trade.above1year	287
de-gaap-ci:bs.ass.currass.receive.other.upto1year	254
de-gaap-ci:bs.eqliab.liab.bank.above1year	183
de-gaap-ci:bs.eqliab.liab.other.above1year	326
de-gaap-ci:bs.eqliab.equity.profsharing	97
de-gaap-ci:bs.eqliab.pretaxres.specamort	3
de-gaap-ci:bs.eqliab.liab.other.above1year.above5years	11
de-gaap-ci:bs.eqliab.liab.advpaym.above1year	47
de-gaap-ci:bs.eqliab.liab.other.above1year.upto5years	3
de-gaap-ci:is.netincome.regular.operatingtc.staff.social.socexp	76
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.finished	47
de-gaap-ci:bs.ass.currass.receive.affil.above1year	59
de-gaap-ci:bs.ass.currass.receive.particip.above1year	24
de-gaap-ci:bs.ass.currass.receive.affil.upto1year	59
de-gaap-ci:bs.eqliab.liab.assoccomp.above1year	188
de-gaap-ci:bs.ass.currass.receive.particip.upto1year	23
de-gaap-ci:bs.ass.fixass.tan.landbuildings.buildingsonownland	50
de-gaap-ci:bs.eqliab.liab.other.shareholders	16
de-gaap-ci:bs.ass.unpaidcap	60
de-gaap-ci:bs.eqliab.equity.revenueres.sharesparentcomp	11
de-gaap-ci:bs.eqliab.equity.minorityinterest	124
de-gaap-ci:bs.eqliab.otherspecres.consolsurplus	14
de-gaap-ci:bs.ass.currass.inventory.allowanceaccounted	1
de-gaap-ci:bs.eqliab.liab.particip.above1year	49
de-gaap-ci:bs.eqliab.liab.particip.upto1year	66
de-gaap-ci:bs.eqliab.liab.securities.convertible	30
de-gaap-ci:bs.eqliab.liab.shareholders.above1year	34
de-gaap-ci:bs.eqliab.liab.shareholders.upto1year	52
de-gaap-ci:incomeuse.gainloss.other	1
de-gaap-ci:incomeuse.gainloss.releaseapreserves	15
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses.regularinterest	46
de-gaap-ci:bs.ass.fixass.intan.concessionbrands.licenses	43
de-gaap-ci:bs.ass.fixass.tan.landbuildings.landwithoutbuildings	16

de-gaap-ci:bs.ass.fixass.tan.inconstradvpaym.advpaym	80
de-gaap-ci:bs.ass.accountingconvenience	41
de-gaap-ci:bs.ass.accountingconvenience.startupcost	43
de-gaap-ci:bs.ass.currass.cashequiv.cash	29
de-gaap-ci:bs.ass.fixass.tan.inconstradvpaym.equipmunderconstr	26
de-gaap-ci:bs.ass.currass.receive.trade.above1year	64
de-gaap-ci:bs.ass.currass.receive.trade.upto1year	65
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses.valuediscount	51
de-gaap-ci:incomeuse.gainloss.releasecapital	9
de-gaap-ci:bs.ass.fixass.intan.concessionbrands.concession	46
de-gaap-ci:bs.eqliab.equity.subscribed.unlimitedliablepartners	86
de-gaap-ci:bs.eqliab.equity.subscribed.unlimitedliablepartners.fixed	4
de-gaap-ci:bs.eqliab.equity.subscribed.limitedliablepartners.fixed	27
de-gaap-ci:is.netincome.regular.operatingcogs	41
de-gaap-ci:is.netincome.regular.operatingcogs.admincost	36
de-gaap-ci:is.netincome.regular.operatingcogs.grossoprofit	40
de-gaap-ci:is.netincome.regular.operatingcogs.grossoprofit.cogs	39
de-gaap-ci:is.netincome.regular.operatingcogs.grossoprofit.netsales	40
de-gaap-ci:is.netincome.regular.operatingcogs.grossoprofit.otherrevenue	41
de-gaap-ci:is.netincome.regular.operatingcogs.othercost	41
de-gaap-ci:is.netincome.regular.operatingcogs.salecost	34
de-gaap-ci:bs.eqliab.equity.silentpartner	20
de-gaap-ci:bs.ass.currass.receive.affil.parentcomp	22
de-gaap-ci:bs.eqliab.liab.notes	60
de-gaap-ci:bs.ass.currass.securities.affil	44
de-gaap-ci:bs.ass.currass.securities.other.other	89
de-gaap-ci:bs.eqliab.equity.profitloss.retainedearnings	131
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .netsales.grosssales	1
de-gaap-ci:is.netincome.regular.fin.netparticipation.earningprofsharing. profpooling	3
de-gaap-ci:bs.ass.taxbalancegenerally	11
de-gaap-ci:bs.eqliab.otherspecres.equitysilentpartner	38
de-gaap-ci:bs.eqliab.equity.revenueres.statutory	39
de-gaap-ci:incomeuse.gainloss.additionotherres	18
de-gaap-ci:is.netincome.regular.operatingcogs.grossoprofit.otherrevenue .exchange	5
de-gaap-ci:is.netincome.regular.operatingcogs.othercost.exchange	5

de-gaap-ci:bs.ass.currass.receiv.shareholders.above1year	4
de-gaap-ci:bs.ass.currass.receiv.shareholders.upto1year	4
de-gaap-ci:bs.eqliab.liab.assoccomp.upto1year.other.parentcomp	9
de-gaap-ci:bs.ass.prepaidexp.loadredempt	31
de-gaap-ci:bs.eqliab.equity.paymforcapitalincrease	14
de-gaap-ci:bs.eqliab.liab.bank.above1year.above5years	7
de-gaap-ci:bs.ass.currass.inventory.inprogress.constructioninprogress	5
de-gaap-ci:bs.ass.other	8
de-gaap-ci:bs.ass.fixass.fin.particip.assoc	17
de-gaap-ci:bs.ass.fixass.fin.particip.other	13
de-gaap-ci:is.netincome.regular.fin.netparticipation.earnings.particip	2
de-gaap-ci:is.netincome.regular.fin.netparticipation.earnings.particip.assoc	2
de-gaap-ci:incomeuse.gainloss.additionpartnersaccount	9
de-gaap-ci:bs.eqliab.equity.revenueres.frompriorperiod	1
de-gaap-ci:bs.eqliab.equity.revenueres.legal.profitformeryear	1
de-gaap-ci:incomeuse.gainloss.releasecapitalreserve	5
de-gaap-ci:bs.eqliab.equity.subscribed.unpaidcap	8
de-gaap-ci:bs.ass.fixass.tan.other	55
de-gaap-ci:is.netincome.extraord.income.other	9
de-gaap-ci:bs.eqliab.liab.securities.upto1year	28
de-gaap-ci:bs.ass.fixass.fin.securities.shares	26
de-gaap-ci:bs.eqliab.equity.revenueres.currchange	4
de-gaap-ci:bs.eqliab.equity.revenueres.special	6
de-gaap-ci:bs.eqliab.liab.securities.above1year	19
de-gaap-ci:bs.eqliab.accruals.pensions.upto1year	2
de-gaap-ci:bs.ass.fixass.tan.otherequipm.factory	8
de-gaap-ci:bs.ass.fixass.intan.selfmade	35
de-gaap-ci:incomeuse.gainloss.additionrevenreserves.legalres	18
de-gaap-ci:bs.ass.fixass.fin.otherfinass.coopshares	47
de-gaap-ci:bs.eqliab.equity.subscribed.cooppartners	9
de-gaap-ci:bs.eqliab.equity.subscribed.cooppartners.staying	5
de-gaap-ci:bs.eqliab.equity.subscribed.cooppartners.leaving	5
de-gaap-ci:bs.eqliab.equity.subscribed.cooppartners.cancelledshares	5
de-gaap-ci:bs.eqliab.equity.revenueres.othercoop	6
de-gaap-ci:bs.ass.unpaidcap.called	2
de-gaap-ci:is.netincome.regular.fin.netparticipation.amortfinanc.seccurrass	14
de-gaap-ci:bs.ass.fixass.tan.other.leasedass	8
de-gaap-ci:is.netincome.regular.fin.netparticipation.earningsecurities.nonassoc	4

de-gaap-ci:bs.ass.currass.receive.affil.trade	18
de-gaap-ci:bs.ass.currass.receive.particip.trade	6
de-gaap-ci:bs.eqliab.liab.assoccomp.upto1year.trade	7
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .material.additpurchc	2
de-gaap-ci:bs.ass.fixass.fin.loanstosharehold	6
de-gaap-ci:bs.ass.currass.inventory.inprogress.ordersinprogress	9
de-gaap-ci:is.netincome.regular.fin.netinterest.income.discount	12
de-gaap-ci:is.netincome.regular.operatingtc.othercost.miscellaneous	1
de-gaap-ci:is.netincome.regular.operatingtc.othercost.insurance	6
de-gaap-ci:bs.ass.fixass.tan.branche_kfz	30
de-gaap-ci:bs.eqliab.equity.capres.other	4
de-gaap-ci:bs.ass.fixass.tan.machinery.installations	6
de-gaap-ci:bs.eqliab.otherspecres.accountingconvenience	1
de-gaap-ci:bs.eqliab.liab.bank.above1year.upto5years	6
de-gaap-ci:bs.ass.fixass.intan.goodwill.fromconsolidation	1
de-gaap-ci:bs.ass.currass.receive.other.othertaxrec	4
de-gaap-ci:bs.eqliab.equity.subscribed.limitedliablepartners.accumloss	2
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue.miscellaneous	6
de-gaap-ci:is.netincome.regular.operatingtc.othercost.marketing	5
de-gaap-ci:is.netincome.regular.operatingtc.othercost.otherordinary	14
de-gaap-ci:is.netincome.regular.operatingtc.othercost.disposcurrass	6
de-gaap-ci:is.netincome.regular.operatingtc.othercost.leasefix	5
de-gaap-ci:is.netincome.regular.operatingtc.othercost.freight	2
de-gaap-ci:incomeuse.gainloss.releaseevenreserves.ownsharesres	1
de-gaap-ci:is.netincome.incomesharing.gain.incomestatement	15
de-gaap-ci:is.netincome.tax.prevperiodreceived	9
de-gaap-ci:bs.ass.currass.inventory.material.consumables	14
de-gaap-ci:bs.eqliab.equity.subscribed.privateaccountsp	1
de-gaap-ci:bs.eqliab.equity.subscribed.privateaccountsp.incomeusedeposits	1
de-gaap-ci:bs.ass.fixass.fin.loanstoparticip.subsidiaries	4
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.losslimitedliablepartner	12
de-gaap-ci:bs.ass.currass.inventory.material.rawmaterial	5
de-gaap-ci:is.netincome.regular.fin.netinterest.income.deposits	1
de-gaap-ci:bs.eqliab.accruals.other.other	5
de-gaap-ci:is.netincome.extraord.expenses.eghgb	9
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.increaseg	3

de-gaap-ci:nt	4
de-gaap-ci:bs.ass.currass.securities.other.securities	3
de-gaap-ci:incomeuse.dividends	2
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.profitloss.showndebit	6
de-gaap-ci:bs.eqliab.equity.subscribed.limitedliablepartners.variable	3
de-gaap-ci:is.netincome.regular.operatingtc.othercost.disposfixass	3
de-gaap-ci:is.netincome.regular.operatingtc.othercost.fixing	2
de-gaap-ci:is.netincome.regular.operatingtc.othercost.vehicles	2
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue.releaseprov	3
de-gaap-ci:bs.contingliab.guaranteesnotescheques	3
de-gaap-ci:bs.eqliab.equity.subscribed.corp.premium	4
de-gaap-ci:is.netincome.regular.operatingtc.othercost.group	1
de-gaap-ci:bs.eqliab.liab.notes.upto1year	3
de-gaap-ci:bs.eqliab.liab.notes.above1year	3
de-gaap-ci:is.netincome.regular.operatingtc.staff.social.welfare	2
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.unlimitedliablepartners.fixed	3
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue.releasepretaxres	1
de-gaap-ci:bs.ass.currass.inventory.finishedandmerch.notyetinvoiced	1
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses.other	19
de-gaap-ci:bs.ass.fixass.fin.particip.silent	14
de-gaap-ci:bs.eqliab.equity.revenueres.forrepaymtocoop	2
de-gaap-ci:bs.eqliab.equity.duetopartners	4
de-gaap-ci:is.netincome.regular.operatingtc.depramort.currass.receiv	1
de-gaap-ci:bs.eqliab.liab.above5years	2
de-gaap-ci:bs.ass.currass.receiv.other.shareholders	2
de-gaap-ci:is.netincome.tax.deferred	3
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .netsales.taxfromgrosssale	1
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.workinpro	2
de-gaap-ci:bs.ass.fixass.fin.securities.other	2
de-gaap-ci:is.netincome.regular.operatingtc.depramort.fixass.startupcost	1
de-gaap-ci:bs.ass.currass.receiv.regulatory	2
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses.discount	5
de-gaap-ci:bs.eqliab.liab.assoccomp.collateralised	2
de-gaap-ci:bs.eqliab.liab.assoccomp.above1year.above5years	3
de-gaap-ci:bs.eqliab.liab.assoccomp.above1year.upto5years	1
de-gaap-ci:is.netincome.regular.fin.netparticipation.amortfinanc.group	2

de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .netsales.group	1
de-gaap-ci:bs.eqliab.equity.subscribed.unlimitedliablepartners.variable	2
de-gaap-ci:is.netincome.incomesharing.loss.other	1
de-gaap-ci:bs.ass.currass.inventory.inprogress.notyetinvoiced	2
de-gaap-ci:incomeuse.gainloss.additionretainedearnings	1
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue.disposfixass	1
de-gaap-ci:bs.eqliab.otherspecres.other	1
de-gaap-ci:incomeuse.gainloss.additionrevenreserves.statres	1
de-gaap-ci:bs.ass.taxbalanceorgancomp	1
de-gaap-ci:bs.ass.fixass.tan.stayingwood	1
de-gaap-ci:incomeuse.gainloss.releasepartnersaccount	2
de-gaap-ci:bs.eqliab.equity.profitloss.tobepaidout	2
de-gaap-ci:bs.ass.fixass.fin.otherloans.other	2
de-gaap-ci:incomeuse.withdrawals	1
de-gaap-ci:bs.ass.currass.cashequiv.chèques	2
de-gaap-ci:bs.eqliab.liab.securities.above1year.above5years	1
de-gaap-ci:bs.eqliab.liab.other.thereoffcoopertiverefunds	1
de-gaap-ci:is.netincome.regular.fin.netinterest.income.other	8
de-gaap-ci:incomeuse.gainloss.releaserevenreserves.legalres	2
de-gaap-ci:mgmtrep	1
de-gaap-ci:bs.eqliab.liab.regulatory	1
de-gaap-ci:bs.ass.currass.receiv.particip.shareholders	1
de-gaap-ci:bs.eqliab.liab.particip.upto1year.shareholders	1
de-gaap-ci:bs.eqliab.liab.bank.collateralised	1
de-gaap-ci:incomeuse.gainloss.minorityint	1
de-gaap-ci:is.netincome.regular.fin.netinterest.expenses.loanfees	1
de-gaap-ci:bs.eqliab.liab.shareholders.upto1year.limitedliable	1
de-gaap-ci:bs.ass.currass.cashequiv.centralbank	1
de-gaap-ci:bs.ass.fixass.fin.loanstoparticip.parentcomp	1
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.withdrawalsunlimitedliablepartner	1
de-gaap-ci:bs.ass.currass.receiv.trade.shareholders	1
de-gaap-ci:bs.ass.currass.inventory.material.supplmaterial	1
de-gaap-ci:bs.ass.prepaidexp.vat	1
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.lossunlimitedliablepartner	1
de-gaap-ci:bs.ass.currass.receiv.shareholder	5
de-gaap-ci:bs.ass.currass.receiv.shareholders.affilcompanies	7
de-gaap-ci:bs.eqliab.liab.shareholders.affilcomp	9

de-gaap-ci:bs.ass.currass.receiv.affil.partner	13
de-gaap-ci:bs.eqliab.liab.assoccomp.partner	17
de-gaap-ci:bs.eqliab.liab.ofwhich toshareholders	6
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.lossunlimitedliablepartners	4
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.privateaccountsp	1
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.privateaccountsp.incomeuseddeposits	1
de-gaap-ci:bs.eqliab.liab.other.upto1year.socsec	1
de-gaap-ci:bs.eqliab.liab.particip.upto1year.trade	3
de-gaap-ci:bs.eqliab.liab.shareholders.upto1year.trade	1
de-gaap-ci:incomeuse.paidincapital.toreserves	2
de-gaap-ci:bs.eqliab.liab.shareholders.unlimitedpartner	3
de-gaap-ci:is.netincome.extraord.income.merger	1
de-gaap-ci:is.netincome.regular.operatingtc.otheroprevenue.other	1
de-gaap-ci:bs.eqliab.liab.above1year	2
de-gaap-ci:bs.eqliab.otherspecres.taxbalancegenerally	2
de-gaap-ci:is.netincome.extraord.expenses.other	1
de-gaap-ci:bs.ass.deficitnotcoveredbycapital.unpaidcapt	1
de-gaap-ci:bs.ass.fixass.tan.branche_kfz.compcar	3
de-gaap-ci:bs.eqliab.liab.assoccomp.upto1year.other	1
de-gaap-ci:bs.eqliab.liab.shareholders.gmbhsilent	1

C.2. XBRL entities not resolvable

Table C.2.: XBRL entities not resolvable using applied taxonomies

de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .material.rawmatconss
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.inprogres
de-gaap-ci:is.netincome.regular.operatingtc.depramort.fixass.intan
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.materialservices .material.additpurchc
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.increaseg
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput.netsales .taxfromgrosssale
de-gaap-ci:is.netincome.regular.operatingtc.grosstradingprofit.totaloutput .inventorychange.workinpro
de-gaap-ci:bs.ass.fixass.tan.stayingwood

D. Algorithmic annotation

D.1. Performance evaluation

Table D.1.: Performance evaluation - annotation of financial entities with filtered expected entity list and standard/customized entity label dictionary - correct entity prediction

Min digits	Label file	Min similarity	TP	TN	FP	FN	Precision	Recall
1	standard	0.3	201	3585	212	151	0.49	0.57
1	standard	0.4	199	3608	190	152	0.51	0.57
1	standard	0.5	199	3630	165	155	0.55	0.56
1	standard	0.6	197	3657	132	163	0.60	0.55
1	standard	0.7	190	3694	86	179	0.67	0.51
1	standard	0.8	178	3731	42	198	0.81	0.47
1	standard	0.9	159	3751	21	218	0.83	0.42
1	standard	1.0	109	3770	2	268	0.98	0.29
2	standard	0.3	202	3603	194	150	0.51	0.57
2	standard	0.4	200	3626	172	151	0.54	0.60
2	standard	0.5	200	3643	152	154	0.57	0.56
2	standard	0.6	198	3670	119	162	0.62	0.55
2	standard	0.7	190	3704	76	179	0.71	0.51
2	standard	0.8	178	3731	42	198	0.81	0.47
2	standard	0.9	159	3751	21	218	0.88	0.42
2	standard	1.0	109	3770	2	268	0.98	0.29
3	standard	0.3	184	3664	117	184	0.61	0.50
3	standard	0.4	181	3678	104	186	0.64	0.49
3	standard	0.5	181	3691	91	186	0.67	0.49
3	standard	0.6	178	3703	76	192	0.70	0.48
3	standard	0.7	172	3727	48	202	0.78	0.46
3	standard	0.8	159	3740	33	217	0.83	0.42
3	standard	0.9	145	3759	13	232	0.92	0.38
3	standard	1.0	99	3770	2	278	0.98	0.26
4	standard	0.3	116	3701	75	257	0.61	0.31

Table D.1.: Performance evaluation - annotation of financial entities with filtered expected entity list and standard/customized entity label dictionary - correct entity prediction

Min digits	Label file	Min similarity	TP	TN	FP	FN	Precision	Recall
4	standard	0.4	113	3714	62	260	0.65	0.31
4	standard	0.5	112	3723	53	261	0.68	0.30
4	standard	0.6	110	3729	46	264	0.71	0.29
4	standard	0.7	105	3744	30	270	0.78	0.28
4	standard	0.8	97	3750	22	280	0.81	0.26
4	standard	0.9	84	3763	9	293	0.90	0.22
4	standard	1.0	64	3770	2	313	0.97	0.17
5	standard	0.3	47	3742	33	327	0.59	0.13
5	standard	0.4	44	3752	23	330	0.66	0.12
5	standard	0.5	44	3757	18	330	0.71	0.12
5	standard	0.6	42	3757	18	332	0.7	0.12
5	standard	0.7	41	3759	15	334	0.73	0.11
5	standard	0.8	37	3764	8	340	0.82	0.10
5	standard	0.9	33	3764	8	344	0.80	0.09
5	standard	1.0	27	3771	1	350	0.96	0.07
1	customized	0.3	209	3632	168	140	0.55	0.59
1	customized	0.4	210	3644	155	140	0.57	0.60
1	customized	0.5	211	3655	143	140	0.60	0.60
1	customized	0.6	213	3676	117	143	0.65	0.60
1	customized	0.7	207	3709	76	157	0.73	0.57
1	customized	0.8	194	3735	44	176	0.82	0.52
1	customized	0.9	179	3746	28	196	0.86	0.48
1	customized	1.0	119	3770	4	256	0.97	0.34
2	customized	0.3	210	3638	161	140	0.57	0.60
2	customized	0.4	211	3650	148	140	0.59	0.60
2	customized	0.5	212	3659	138	140	0.61	0.60
2	customized	0.6	214	3680	112	143	0.67	0.60
2	customized	0.7	208	3710	74	157	0.74	0.60
2	customized	0.8	194	3725	44	176	0.82	0.52
2	customized	0.9	179	3746	28	196	0.87	0.48
2	customized	1.0	119	3770	4	256	0.97	0.32
3	customized	0.3	192	3693	88	176	0.69	0.52
3	customized	0.4	192	3696	85	176	0.69	0.52
3	customized	0.5	193	3703	78	175	0.71	0.52
3	customized	0.6	193	3710	69	177	0.74	0.52

Table D.1.: Performance evaluation - annotation of financial entities with filtered expected entity list and standard/customized entity label dictionary - correct entity prediction

Min digits	Label file	Min similarity	TP	TN	FP	FN	Precision	Recall
3	customized	0.7	185	3733	44	187	0.81	0.50
3	customized	0.8	172	3744	32	201	0.84	0.46
3	customized	0.9	159	3754	20	216	0.89	0.42
3	customized	1.0	105	3770	4	270	0.96	0.28
4	customized	0.3	132	3722	54	241	0.71	0.35
4	customized	0.4	132	3724	52	241	0.71	0.35
4	customized	0.5	132	3730	46	241	0.74	0.35
4	customized	0.6	130	3733	42	244	0.76	0.35
4	customized	0.7	125	3749	25	250	0.83	0.33
4	customized	0.8	119	3755	19	256	0.86	0.32
4	customized	0.9	107	3760	14	268	0.88	0.26
4	customized	1.0	79	3770	4	296	0.95	0.21

Table D.2.: Performance evaluation - annotation of financial entities with filtered expected entity list, customized entity label dictionary and multi-match barrier - correct entity prediction

Min digits	Min similarity	TP	TN	FP	FN	Precision	Recall
1	0,10	141	302	80	41	0,64	0,77
1	0,20	141	302	80	41	0,64	0,77
1	0,30	139	302	81	42	0,63	0,77
1	0,40	140	305	78	41	0,64	0,77
1	0,50	141	310	75	38	0,65	0,79
1	0,60	144	319	62	39	0,70	0,79
1	0,70	133	133	44	54	0,75	0,71
1	0,80	122	349	26	67	0,82	0,65
1	0,90	109	352	22	81	0,83	0,57
1	1,00	81	364	7	112	0,92	0,42
2	0,10	143	309	73	39	0,66	0,79
2	0,20	143	309	73	39	0,66	0,79
2	0,30	141	309	74	40	0,66	0,78
2	0,40	142	312	71	39	0,67	0,78
2	0,50	142	316	69	37	0,67	0,79
2	0,60	145	326	55	38	0,73	0,79
2	0,70	134	336	41	53	0,77	0,72

Table D.2.: Performance evaluation - annotation of financial entities with filtered expected entity list, customized entity label dictionary and multi-match barrier - correct entity prediction

Min digits	Min similarity	TP	TN	FP	FN	Precision	Recall
2	0,80	122	349	26	67	0,82	0,65
2	0,90	109	352	22	81	0,83	0,57
2	1,00	81	364	7	112	0,92	0,42
3	0,10	143	345	32	44	0,82	0,76
3	0,20	143	345	32	44	0,82	0,76
3	0,30	141	345	32	46	0,82	0,75
3	0,40	139	346	31	48	0,82	0,74
3	0,50	139	348	29	48	0,83	0,74
3	0,60	138	352	24	50	0,85	0,73
3	0,70	126	357	18	63	0,88	0,67
3	0,80	114	361	14	75	0,89	0,60
3	0,90	102	362	12	88	0,89	0,54
3	1,00	77	366	5	116	0,94	0,40
4	0,10	101	356	17	90	0,86	0,53
4	0,20	101	356	17	90	0,86	0,53
4	0,30	100	356	17	91	0,85	0,52
4	0,40	99	357	16	92	0,86	0,52
4	0,50	98	358	15	93	0,87	0,51
4	0,60	94	360	13	97	0,88	0,49
4	0,70	85	363	10	106	0,89	0,45
4	0,80	79	364	9	112	0,90	0,41
4	0,90	72	364	8	120	0,90	0,38
4	1,00	57	367	4	136	0,93	0,30
5	0,10	57	364	7	136	0,89	0,30
5	0,20	57	364	7	136	0,89	0,30
5	0,30	56	364	7	137	0,89	0,29
5	0,40	56	364	7	137	0,89	0,29
5	0,50	55	364	7	138	0,89	0,28
5	0,60	52	364	7	141	0,88	0,27
5	0,70	47	365	6	146	0,89	0,24
5	0,80	44	366	5	149	0,90	0,23
5	0,90	42	366	5	151	0,89	0,22
5	1,00	33	367	4	160	0,89	0,17

Table D.3.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using FastText word embeddings - correct entity prediction

Detection type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Value	1	0,1	144	292	90	38	0,62	0,79
Fin. Value	1	0,2	144	292	90	38	0,62	0,79
Fin. Value	1	0,3	143	292	90	39	0,61	0,79
Fin. Value	1	0,4	140	291	91	42	0,61	0,77
Fin. Value	1	0,5	141	302	78	43	0,64	0,77
Fin. Value	1	0,6	130	339	38	57	0,77	0,70
Fin. Value	1	0,7	118	363	10	73	0,92	0,62
Fin. Value	1	0,8	104	364	8	88	0,93	0,54
Fin. Value	1	0,9	88	364	8	104	0,92	0,46
Fin. Value	1	1	81	364	8	111	0,91	0,42
Fin. Value	2	0,1	149	309	73	33	0,67	0,82
Fin. Value	2	0,2	149	309	73	33	0,67	0,82
Fin. Value	2	0,3	148	309	73	34	0,67	0,81
Fin. Value	2	0,4	145	308	74	37	0,66	0,80
Fin. Value	2	0,5	146	317	63	38	0,70	0,79
Fin. Value	2	0,6	134	349	28	53	0,83	0,72
Fin. Value	2	0,7	118	363	10	73	0,92	0,62
Fin. Value	2	0,8	104	364	8	88	0,93	0,54
Fin. Value	2	0,9	88	364	8	104	0,92	0,46
Fin. Value	2	1	81	364	8	111	0,91	0,42
Fin. Value	3	0,1	146	345	33	40	0,82	0,78
Fin. Value	3	0,2	146	345	33	40	0,82	0,78
Fin. Value	3	0,3	146	345	33	40	0,82	0,78
Fin. Value	3	0,4	144	345	33	42	0,81	0,77
Fin. Value	3	0,5	142	348	28	46	0,84	0,76
Fin. Value	3	0,6	128	360	14	62	0,90	0,67
Fin. Value	3	0,7	110	365	8	81	0,93	0,58
Fin. Value	3	0,8	96	366	6	96	0,94	0,50
Fin. Value	3	0,9	84	366	6	108	0,93	0,44
Fin. Value	3	1	77	366	6	115	0,93	0,40
Fin. Value	4	0,1	103	356	18	87	0,85	0,54
Fin. Value	4	0,2	103	356	18	87	0,85	0,54
Fin. Value	4	0,3	103	356	18	87	0,85	0,54
Fin. Value	4	0,4	102	356	18	88	0,85	0,54
Fin. Value	4	0,5	99	358	15	92	0,87	0,52

Table D.3.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using FastText word embeddings - correct entity prediction

Detection type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Value	4	0,6	89	365	7	103	0,93	0,46
Fin. Value	4	0,7	77	366	6	115	0,93	0,40
Fin. Value	4	0,8	65	367	4	128	0,94	0,34
Fin. Value	4	0,9	61	367	4	132	0,94	0,32
Fin. Value	4	1	57	367	4	136	0,93	0,30
Fin. Value	5	0,1	54	364	7	139	0,89	0,28
Fin. Value	5	0,2	54	364	7	139	0,89	0,28
Fin. Value	5	0,3	54	364	7	139	0,89	0,28
Fin. Value	5	0,4	54	364	7	139	0,89	0,28
Fin. Value	5	0,5	53	365	6	140	0,90	0,27
Fin. Value	5	0,6	48	366	5	145	0,91	0,25
Fin. Value	5	0,7	41	366	5	152	0,89	0,21
Fin. Value	5	0,8	36	367	4	157	0,90	0,19
Fin. Value	5	0,9	35	367	4	158	0,90	0,18
Fin. Value	5	1	33	367	4	160	0,89	0,17
Fin. Value	6	0,1	24	365	5	170	0,83	0,12
Fin. Value	6	0,2	24	365	5	170	0,83	0,12
Fin. Value	6	0,3	24	365	5	170	0,83	0,12
Fin. Value	6	0,4	23	365	5	171	0,82	0,12
Fin. Value	6	0,5	23	366	4	171	0,85	0,12
Fin. Value	6	0,6	20	366	4	174	0,83	0,10
Fin. Value	6	0,7	18	366	4	176	0,82	0,09
Fin. Value	6	0,8	16	367	3	178	0,84	0,08
Fin. Value	6	0,9	16	367	3	178	0,84	0,08
Fin. Value	6	1	16	367	3	178	0,84	0,08
Fin. Entity	1	0,1	211	3459	347	132	0,38	0,62
Fin. Entity	1	0,2	211	3459	347	132	0,38	0,62
Fin. Entity	1	0,3	211	3460	346	132	0,38	0,62
Fin. Entity	1	0,4	211	3462	344	132	0,38	0,62
Fin. Entity	1	0,5	215	3486	320	128	0,40	0,63
Fin. Entity	1	0,6	218	3631	162	138	0,57	0,61
Fin. Entity	1	0,7	211	3748	27	163	0,89	0,56
Fin. Entity	1	0,8	183	3758	17	191	0,92	0,49
Fin. Entity	1	0,9	146	3765	10	228	0,94	0,39
Fin. Entity	1	1	131	3770	5	243	0,96	0,35

Table D.3.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using FastText word embeddings - correct entity prediction

Detection type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Entity	2	0,1	218	3501	299	131	0,42	0,62
Fin. Entity	2	0,2	218	3501	299	131	0,42	0,62
Fin. Entity	2	0,3	218	3502	298	131	0,42	0,62
Fin. Entity	2	0,4	218	3504	296	131	0,42	0,62
Fin. Entity	2	0,5	222	3527	273	127	0,45	0,64
Fin. Entity	2	0,6	222	3661	126	140	0,64	0,61
Fin. Entity	2	0,7	211	3748	27	163	0,89	0,56
Fin. Entity	2	0,8	183	3758	17	191	0,92	0,49
Fin. Entity	2	0,9	146	3765	10	228	0,94	0,39
Fin. Entity	2	1	131	3770	5	243	0,96	0,35
Fin. Entity	3	0,1	209	3595	187	158	0,53	0,57
Fin. Entity	3	0,2	209	3595	187	158	0,53	0,57
Fin. Entity	3	0,3	210	3596	186	157	0,53	0,57
Fin. Entity	3	0,4	210	3598	184	157	0,53	0,57
Fin. Entity	3	0,5	211	3612	169	157	0,56	0,57
Fin. Entity	3	0,6	206	3698	80	165	0,72	0,56
Fin. Entity	3	0,7	190	3753	22	184	0,90	0,51
Fin. Entity	3	0,8	161	3760	15	213	0,91	0,43
Fin. Entity	3	0,9	132	3765	10	242	0,93	0,35
Fin. Entity	3	1	117	3770	5	257	0,96	0,31
Fin. Entity	4	0,1	130	3657	120	242	0,52	0,35
Fin. Entity	4	0,2	130	3657	120	242	0,52	0,35
Fin. Entity	4	0,3	131	3658	119	241	0,52	0,35
Fin. Entity	4	0,4	132	3660	117	240	0,53	0,35
Fin. Entity	4	0,5	131	3669	107	242	0,55	0,35
Fin. Entity	4	0,6	128	3729	46	246	0,74	0,34
Fin. Entity	4	0,7	118	3758	16	257	0,88	0,31
Fin. Entity	4	0,8	97	3765	9	278	0,92	0,26
Fin. Entity	4	0,9	86	3765	9	289	0,91	0,23
Fin. Entity	4	1	79	3770	4	296	0,95	0,21
Fin. Entity	5	0,1	53	3729	46	321	0,54	0,14
Fin. Entity	5	0,2	53	3729	46	321	0,54	0,14
Fin. Entity	5	0,3	53	3729	46	321	0,54	0,14
Fin. Entity	5	0,4	54	3730	45	320	0,55	0,14
Fin. Entity	5	0,5	54	3736	39	320	0,58	0,14

Table D.3.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using FastText word embeddings - correct entity prediction

Detection type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Entity	5	0,6	52	3748	26	323	0,67	0,14
Fin. Entity	5	0,7	48	3766	8	327	0,86	0,13
Fin. Entity	5	0,8	39	3766	8	336	0,83	0,10
Fin. Entity	5	0,9	37	3766	8	338	0,82	0,10
Fin. Entity	5	1	35	3771	3	340	0,92	0,09
Fin. Entity	6	0,1	22	3756	17	354	0,56	0,06
Fin. Entity	6	0,2	22	3756	17	354	0,56	0,06
Fin. Entity	6	0,3	22	3756	17	354	0,56	0,06
Fin. Entity	6	0,4	22	3757	16	354	0,58	0,06
Fin. Entity	6	0,5	22	3758	15	354	0,59	0,06
Fin. Entity	6	0,6	20	3766	7	356	0,74	0,05
Fin. Entity	6	0,7	19	3771	2	357	0,90	0,05
Fin. Entity	6	0,8	15	3771	2	361	0,88	0,04
Fin. Entity	6	0,9	15	3771	2	361	0,88	0,04
Fin. Entity	6	1	15	3771	2	361	0,88	0,04

Table D.4.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using Levenshtein distance similarity - correct entity prediction

Detection Type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Value	1	0,1	153	293	92	26	0,62	0,85
Fin. Value	1	0,2	156	292	95	21	0,62	0,88
Fin. Value	1	0,3	151	304	80	29	0,65	0,84
Fin. Value	1	0,4	133	347	28	56	0,83	0,70
Fin. Value	1	0,5	126	361	14	63	0,90	0,67
Fin. Value	1	0,6	121	363	10	70	0,92	0,63
Fin. Value	1	0,7	114	364	8	78	0,93	0,59
Fin. Value	1	0,8	111	364	8	81	0,93	0,58
Fin. Value	1	0,9	108	364	8	84	0,93	0,56
Fin. Value	1	1	81	364	8	111	0,91	0,42
Fin. Value	2	0,1	158	310	75	21	0,68	0,88
Fin. Value	2	0,2	160	308	79	17	0,67	0,90
Fin. Value	2	0,3	153	317	67	27	0,70	0,85
Fin. Value	2	0,4	133	347	28	56	0,83	0,70

Table D.4.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using Levenshtein distance similarity - correct entity prediction

Detection Type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Value	2	0,5	126	361	14	63	0,90	0,67
Fin. Value	2	0,6	121	363	10	70	0,92	0,63
Fin. Value	2	0,7	114	364	8	78	0,93	0,59
Fin. Value	2	0,8	111	364	8	81	0,93	0,58
Fin. Value	2	0,9	108	364	8	84	0,93	0,56
Fin. Value	2	1	81	364	8	111	0,91	0,42
Fin. Value	3	0,1	150	345	33	36	0,82	0,81
Fin. Value	3	0,2	153	345	33	33	0,82	0,82
Fin. Value	3	0,3	144	349	28	43	0,84	0,77
Fin. Value	3	0,4	124	360	15	65	0,89	0,66
Fin. Value	3	0,5	117	364	11	72	0,91	0,62
Fin. Value	3	0,6	113	364	9	78	0,93	0,59
Fin. Value	3	0,7	107	365	7	85	0,94	0,56
Fin. Value	3	0,8	104	365	7	88	0,94	0,54
Fin. Value	3	0,9	102	365	7	90	0,94	0,53
Fin. Value	3	1	77	366	6	115	0,93	0,40
Fin. Value	4	0,1	106	356	18	84	0,85	0,56
Fin. Value	4	0,2	107	356	18	83	0,86	0,56
Fin. Value	4	0,3	100	357	17	90	0,85	0,53
Fin. Value	4	0,4	85	364	9	106	0,90	0,45
Fin. Value	4	0,5	82	366	7	109	0,92	0,43
Fin. Value	4	0,6	78	366	6	114	0,93	0,41
Fin. Value	4	0,7	75	366	5	118	0,94	0,39
Fin. Value	4	0,8	74	366	5	119	0,94	0,38
Fin. Value	4	0,9	72	366	5	121	0,94	0,37
Fin. Value	4	1	57	367	4	136	0,93	0,30
Fin. Value	5	0,1	57	364	7	136	0,89	0,30
Fin. Value	5	0,2	57	364	7	136	0,89	0,30
Fin. Value	5	0,3	52	364	7	141	0,88	0,27
Fin. Value	5	0,4	46	365	6	147	0,88	0,24
Fin. Value	5	0,5	44	366	5	149	0,90	0,23
Fin. Value	5	0,6	43	366	5	150	0,90	0,22
Fin. Value	5	0,7	42	366	5	151	0,89	0,22
Fin. Value	5	0,8	42	366	5	151	0,89	0,22
Fin. Value	5	0,9	41	366	5	152	0,89	0,21

Table D.4.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using Levenshtein distance similarity - correct entity prediction

Detection Type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Value	5	1	33	367	4	160	0,89	0,17
Fin. Entity	1	0,1	223	3549	250	127	0,47	0,64
Fin. Entity	1	0,2	227	3550	251	121	0,47	0,65
Fin. Entity	1	0,3	222	3600	199	128	0,53	0,63
Fin. Entity	1	0,4	222	3719	64	144	0,78	0,61
Fin. Entity	1	0,5	219	3743	34	153	0,87	0,59
Fin. Entity	1	0,6	213	3754	22	160	0,91	0,57
Fin. Entity	1	0,7	200	3758	17	174	0,92	0,53
Fin. Entity	1	0,8	194	3762	13	180	0,94	0,52
Fin. Entity	1	0,9	187	3767	8	187	0,96	0,50
Fin. Entity	1	1	131	3770	5	243	0,96	0,35
Fin. Entity	2	0,1	228	3573	221	127	0,51	0,64
Fin. Entity	2	0,2	231	3572	224	122	0,51	0,65
Fin. Entity	2	0,3	224	3619	177	129	0,56	0,63
Fin. Entity	2	0,4	222	3719	64	144	0,78	0,61
Fin. Entity	2	0,5	219	3743	34	153	0,87	0,59
Fin. Entity	2	0,6	213	3754	22	160	0,91	0,57
Fin. Entity	2	0,7	200	3758	17	174	0,92	0,53
Fin. Entity	2	0,8	194	3762	13	180	0,94	0,52
Fin. Entity	2	0,9	187	3767	8	187	0,96	0,50
Fin. Entity	2	1	131	3770	5	243	0,96	0,35
Fin. Entity	3	0,1	208	3653	133	155	0,61	0,57
Fin. Entity	3	0,2	211	3653	133	152	0,61	0,58
Fin. Entity	3	0,3	203	3676	109	161	0,65	0,56
Fin. Entity	3	0,4	199	3736	45	169	0,82	0,54
Fin. Entity	3	0,5	196	3748	29	176	0,87	0,53
Fin. Entity	3	0,6	192	3756	20	181	0,91	0,51
Fin. Entity	3	0,7	181	3759	16	193	0,92	0,48
Fin. Entity	3	0,8	175	3763	12	199	0,94	0,47
Fin. Entity	3	0,9	170	3767	8	204	0,96	0,45
Fin. Entity	3	1	117	3770	5	257	0,96	0,31
Fin. Entity	4	0,1	137	3699	80	233	0,63	0,37
Fin. Entity	4	0,2	138	3699	80	232	0,63	0,37
Fin. Entity	4	0,3	133	3708	71	237	0,65	0,36
Fin. Entity	4	0,4	129	3751	25	244	0,84	0,35

Table D.4.: Performance evaluation - annotation of financial entities and values with filtered expected entity list, customized entity label dictionary and multi match barrier using Levenshtein distance similarity - correct entity prediction

Detection Type	Min dig.	Min sim.	TP	TN	FP	FN	Precision	Recall
Fin. Entity	4	0,5	128	3756	19	246	0,87	0,34
Fin. Entity	4	0,6	124	3764	11	250	0,92	0,33
Fin. Entity	4	0,7	116	3765	9	259	0,93	0,31
Fin. Entity	4	0,8	114	3766	8	261	0,93	0,30
Fin. Entity	4	0,9	109	3770	4	266	0,96	0,29
Fin. Entity	4	1	79	3770	4	296	0,95	0,21
Fin. Entity	5	0,1	60	3746	29	314	0,67	0,16
Fin. Entity	5	0,2	60	3746	29	314	0,67	0,16
Fin. Entity	5	0,3	56	3752	23	318	0,71	0,15
Fin. Entity	5	0,4	53	3764	11	321	0,83	0,14
Fin. Entity	5	0,5	53	3766	8	322	0,87	0,14
Fin. Entity	5	0,6	52	3767	7	323	0,88	0,14
Fin. Entity	5	0,7	51	3767	7	324	0,88	0,14
Fin. Entity	5	0,8	51	3767	7	324	0,88	0,14
Fin. Entity	5	0,9	50	3771	3	325	0,94	0,13
Fin. Entity	5	1	35	3771	3	340	0,92	0,09

Table D.5.: Final performance evaluation - algorithmic annotation of financial values and financial entities using FastText vectors

Min digits	Min similarity	Precision	Recall	F0.25
1,00	0,10	0,50	0,70	0,51
1,00	0,20	0,50	0,70	0,51
1,00	0,30	0,50	0,70	0,50
1,00	0,40	0,49	0,69	0,50
1,00	0,50	0,52	0,70	0,53
1,00	0,60	0,67	0,65	0,67
1,00	0,70	0,90	0,59	0,88
1,00	0,80	0,92	0,52	0,88
1,00	0,90	0,93	0,42	0,87
1,00	1,00	0,94	0,39	0,86
2,00	0,10	0,55	0,72	0,55
2,00	0,20	0,55	0,72	0,55
2,00	0,30	0,55	0,72	0,55
2,00	0,40	0,54	0,71	0,55

Table D.5.: Final performance evaluation - algorithmic annotation of financial values and financial entities using FastText vectors

Min digits	Min similarity	Precision	Recall	F0.25
2,00	0,50	0,57	0,71	0,58
2,00	0,60	0,73	0,66	0,73
2,00	0,70	0,90	0,59	0,88
2,00	0,80	0,92	0,52	0,88
2,00	0,90	0,93	0,42	0,87
2,00	1,00	0,94	0,39	0,86
3,00	0,10	0,67	0,68	0,67
3,00	0,20	0,67	0,68	0,67
3,00	0,30	0,67	0,68	0,67
3,00	0,40	0,67	0,67	0,67
3,00	0,50	0,70	0,66	0,69
3,00	0,60	0,81	0,61	0,80
3,00	0,70	0,91	0,54	0,88
3,00	0,80	0,93	0,47	0,88
3,00	0,90	0,93	0,40	0,86
3,00	1,00	0,94	0,36	0,86
4,00	0,10	0,69	0,45	0,66
4,00	0,20	0,69	0,45	0,66
4,00	0,30	0,69	0,45	0,67
4,00	0,40	0,69	0,45	0,67
4,00	0,50	0,71	0,43	0,68
4,00	0,60	0,83	0,40	0,78
4,00	0,70	0,90	0,36	0,83
4,00	0,80	0,93	0,30	0,83
4,00	0,90	0,92	0,27	0,81
4,00	1,00	0,94	0,25	0,81
5,00	0,10	0,71	0,21	0,62
5,00	0,20	0,71	0,21	0,62
5,00	0,30	0,71	0,21	0,62
5,00	0,40	0,72	0,21	0,63
5,00	0,50	0,74	0,21	0,64
5,00	0,60	0,79	0,19	0,67
5,00	0,70	0,87	0,17	0,70
5,00	0,80	0,86	0,15	0,67
5,00	0,90	0,86	0,14	0,66

Table D.5.: Final performance evaluation - algorithmic annotation of financial values and financial entities using FastText vectors

Min digits	Min similarity	Precision	Recall	F0.25
5,00	1,00	0,91	0,13	0,67

Table D.6.: Final performance evaluation - algorithmic annotation of financial values and financial entities using Levenshtein distance

Min digits	Min similarity	Precision	Recall	F0.25
1	0,1	0,55	0,75	0,56
1	0,2	0,55	0,77	0,56
1	0,3	0,59	0,74	0,60
1	0,4	0,80	0,66	0,79
1	0,5	0,88	0,63	0,86
1	0,6	0,92	0,60	0,89
1	0,7	0,93	0,56	0,89
1	0,8	0,93	0,55	0,90
1	0,9	0,95	0,53	0,90
1	1	0,94	0,39	0,86
2	0,1	0,59	0,76	0,60
2	0,2	0,59	0,78	0,60
2	0,3	0,63	0,74	0,63
2	0,4	0,80	0,66	0,79
2	0,5	0,88	0,63	0,86
2	0,6	0,92	0,60	0,89
2	0,7	0,93	0,56	0,89
2	0,8	0,93	0,55	0,90
2	0,9	0,95	0,53	0,90
2	1	0,94	0,39	0,86
3	0,1	0,71	0,69	0,71
3	0,2	0,72	0,70	0,72
3	0,3	0,74	0,66	0,74
3	0,4	0,85	0,60	0,83
3	0,5	0,89	0,57	0,86
3	0,6	0,92	0,55	0,88
3	0,7	0,93	0,52	0,89
3	0,8	0,94	0,50	0,89
3	0,9	0,95	0,49	0,90
3	1	0,94	0,36	0,86

Table D.6.: Final performance evaluation - algorithmic annotation of financial values and financial entities using Levenshtein distance

Min digits	Min similarity	Precision	Recall	F0.25
4	0,1	0,74	0,46	0,72
4	0,2	0,74	0,47	0,72
4	0,3	0,75	0,44	0,72
4	0,4	0,87	0,40	0,81
4	0,5	0,90	0,39	0,83
4	0,6	0,92	0,37	0,85
4	0,7	0,93	0,35	0,85
4	0,8	0,94	0,34	0,85
4	0,9	0,95	0,33	0,86
4	1	0,94	0,25	0,81
5	0,1	0,78	0,23	0,68
5	0,2	0,78	0,23	0,68
5	0,3	0,80	0,21	0,68
5	0,4	0,86	0,19	0,71
5	0,5	0,88	0,18	0,72
5	0,6	0,89	0,18	0,72
5	0,7	0,89	0,18	0,72
5	0,8	0,89	0,18	0,72
5	0,9	0,92	0,17	0,73
5	1	0,91	0,13	0,67