T. Grust et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2019),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 149

A Hybrid Information Extraction Approach Exploiting Structured Data within a Text Mining Process

Cornelia Kiefer[1], Peter Reimann[2], Bernhard Mitschang[3]

Abstract: Many data sets encompass structured data fields with embedded free text fields. The text fields allow customers and workers to input information which cannot be encoded in structured fields. Several approaches use structured and unstructured data in isolated analyses. The result of isolated mining of structured data fields misses crucial information encoded in free text. The result of isolated text mining often mainly repeats information already available from structured data. The actual information gain of isolated text mining is thus limited. The main drawback of both isolated approaches is that they may miss crucial information. The hybrid information extraction approach suggested in this paper adresses this issue. Instead of extracting information that in large parts was already available beforehand, it extracts new, valuable information from free texts. Our solution exploits results of analyzing structured data within the text mining process, i.e., structured information guides and improves the information extraction process on textual data. Our main contributions comprise the description of the concept of hybrid information extraction as well as a prototypical implementation and an evaluation with two real-world data sets from aftersales and production with English and German free text fields.

Keywords: information extraction, clustering, text mining, free text fields

1 Introduction

Many data sets in research and industry capture information both in structured and unstructured data fields. Structured data fields are suitable if the data type and value domain fit the perceived purpose. For example, structured data fields are appropriate to store the duration of a downtime in a production line in seconds. Unstructured data fields are better if no suitable structured type is available or if one needs to express certain issues in natural language to be readable and understandable by human users. For example, unstructured free text fields are adequate when explaining how to repair a machine, since this information is complex and cannot be captured in structured data. Especially, humans tend to provide more complete information using natural language texts than using structured

[1] University of Stuttgart, Graduate School of Excellence Advanced Manufacturing Engineering, Nobelstr. 12, Germany cornelia.kiefer@gsame.uni-stuttgart.de
[2] University of Stuttgart, Graduate School of Excellence Advanced Manufacturing Engineering, Nobelstr. 12, Germany peter.reimann@gsame.uni-stuttgart.de
[3] University of Stuttgart, Institute for Parallel and Distributed Systems, Universitätsstraße 38, Germany bernhard.mitschang@ipvs.uni-stuttgart.de

doi:10.18420/btw2019-10

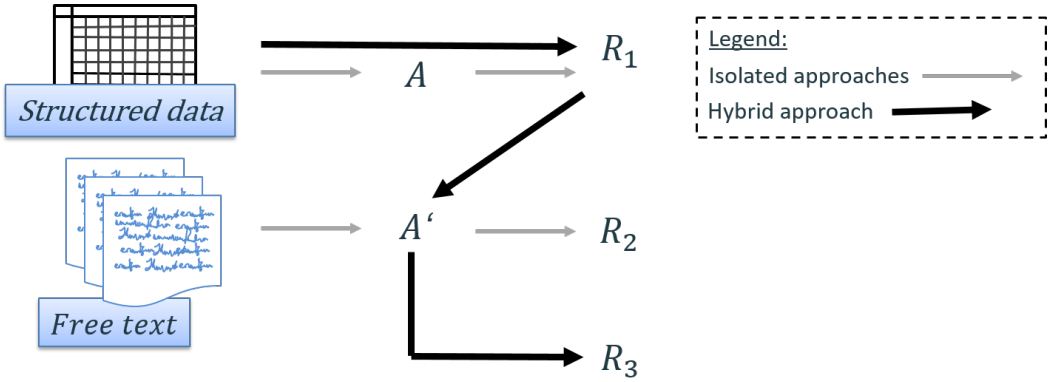

Fig. 1: Isolated information extraction approaches A on structured data and A' on unstructured text data yield results R_1 and R_2. Our hybrid information extraction approach uses analysis result R_1 in the text mining process and yields result R_3, thus extending results R_1 and R_2.

information [HW96]. Thus, it is important to extract information not only from structured data, but also from unstructured, e.g., textual data as is mostly available in data sets from production, aftersales and research.

Standard approaches for information extraction from unstructured text data do not use structured data in text analysis (see Section 2). The result may be redundant information already available from structured data. Moreover, isolated approaches on structured data also miss crucial information. As illustrated in Figure 1, in contrast to these approaches, our hybrid information extraction approach exploits analysis results obtained from structured data in the text analysis pipeline.

The hybrid information extraction concept may be applied to all information extraction approaches on structured data with embedded or linked text data. In this paper, we present the concept and a prototypical implementation. In the evaluation, we present two real-world data sets and apply the method to them. Since for these data sets and use cases, no information on what to extract from the data is known beforehand, and no training data sets are available, we select a clustering information extraction approach for the prototypical implementation. Clustering is a robust and unsupervised means to extract information from text.

The goal of the suggested hybrid approach (yielding R_3 in Figure 1) is to increase the amount of new information, when compared to the information gained by the two isolated approaches (R_1 and R_2 in Figure 1). In our evaluation based on the two real-world data sets and the prototype, we denote the **'degree of new information'** with i_{new} and define it as shown in Formula 1.

$$i_{new} = (c_{new}/N) \tag{1}$$

where c_{new} is the number of cluster names not already known from the structured column and N is the number of all clusters considered.

By employing this purely quantitative metric, we are able to compare $R_1 - R_3$ in a straightforward way. Additionally, we can compare our results to the work of Ghazizadeh et al. [GML14]. In future work, we are going to consider more qualitative insights as well as additional quantitative metrics, e.g., based on entropy.

With an isolated approach on structured data, all information is extracted from the structured data fields only. Thus, based on the definition above, $i_{new} = 0$. Only, if information from free text is employed, i_{new} increases. For example, if half of the cluster names are not yet present in the structured information, i_{new} is 0.5. For R_2 and R_3, i_{new} differs due to the exploitation of structured information available in the hybrid approach (R_3). By filtering out redundant information within the text mining process, the amount of new information increases. For our prototype and the two real-world data sets, for R_3, i_{new} is 0.21 and 0.43 higher, than for R_2.

The main contributions of this paper are:

- A description of the concept of hybrid information extraction.
- A discussion of design issues of a prototypical implementation of the approach for English as well as German free text fields.
- An evaluation of the hybrid information extraction approach. Here, we compare the results of isolated approaches (cf. R_1 and R_2 in Figure 1) with results of our hybrid approach (cf. R_3 in Figure 1) and we show that i_{new} is higher for the hybrid approach. For this purpose, we apply the prototype to an open dataset on problems with cars in aftersales (NHTSA data set[4]) and to a dataset on downtimes in a production line.

In the next two sections, we give an overview on work related to our approach and motivate hybrid information extraction with an example use case. In Section 4, we describe our method used for hybrid information extraction, and we discuss implementation details in Section 5. We illustrate the benefit of our approach with two data sets in Section 6 and conclude in Section 7.

2 Related Work

Plenty research works propose text mining approaches on free text. Compared to our work, these publications make no use of analytical results of structured data in the text mining process. Many approaches look at free text information in isolation. In the following, we present an excerpt of these approaches which work with real data sets: Carter et al. show a use case in the pharmaceutics domain where they mine the Pillreports.com database using the k-means algorithm [CH14]. Gamon et al. apply clustering to mine opinions on cars in

[4] https://www-odi.nhtsa.dot.gov/downloads/

the car reviews database[5]. The approach is based on a self-defined clustering algorithm [Ga05]. Brooks focuses on preventing industrial accidents [Br08]. In his approach, the SAS Text Miner Software is used to mine workers' compensation claims data. Clustering is based on the Expectation Maximization algorithm. Forman et al. assist technical support staff in a call center applying a self-developed clustering method on call logs [FKS06]. In many of the data sets used in these isolated approaches, also information in structured data fields is available. The main drawback of these approaches is however that they do not make use of this information source.

Many approaches use both structured and unstructured information in parallel. Yet, these approaches solely integrate the results of the isolated approaches. For many data sets like the data considered in this work such approaches are problematic, since valuable information may get lost (see Section 3). For example, Tan et al. mined data of a service center to get information on the expected processing times of service requests. Mining is based on structured data (the processing times) and case descriptions in free text fields [Ta00]. In their approach, they build a classification model which uses features induced separately from structured and text data. Chougule et al. speed up repair tasks of cars based on a framework, which combines association rule mining, case-based-reasoning and text mining [CRB11]. While the whole framework considers structured as well as unstructured data, the text mining component analyses the texts in isolation using hierachical clustering algorithms.

Similarly, many approaches convert unstructured text data into structured data fields with the goal of merging structured and converted unstructured data and information. In contrast to our method, the information extraction methods work on the texts in isolation. For example, the DeepDive system structures free texts using statistical inference and machine learning [Ce15]. Gubanov et al. present the data tamer system [GSB14], where the structuring of textual data is based on external tools that are not described in more detail. After the conversion of the unstructured text data, modules such as schema integration and entity consolidation in data tamer may be applied.

Various approaches to information extraction are called hybrid, since they combine two machine learning algorithms. Silva et al. combine naive bayes, the PART algorithm and the k-nearest-neighbour with hidden markov models [SBP06]. Xiao et al. combine maximum entropy and maximum entropy markov models [XZZ08]. These approaches still analyze one type of data in isolation. They are not hybrid in the sense of using analytical results on structured data in the text mining process.

The work most related to ours is by Ghazizadeh et al. [GML14] and uses the same data set as we use (NHTSA data set, see Sections 3 and 6.1). Ghazizadeh et al. investigate reasons for fatal car accidents. They apply latent semantic analysis and hierarchical clustering to the free text fields in isolation. In difference to our approach, this work uses structured information in a first step only, before clustering takes place, to filter out the relevant part of the data. All structured information are ignored in the next steps of the information

[5] https://www.msn.com/en-us/autos/

extraction process. Ghazizadeh et al. [GML14] present evaluation results which show that half of the cluster names correspond to vehicle components. These vehicle components represent information which is also available in a structured data field in the data set. Thus, only half of the clusters represent new and relevant information. In the hybrid approach suggested in this work we adress this inconvenience.

While many approaches for the extraction of information from structured and free text data exist, the main drawback is that they are isolated: they do not employ structured information that is available and helpful within the text mining process. In this paper, we adress this issue and show with two data sets from the product lifecycle that a hybrid approach to information extraction leads to new information that otherwise would be hidden behind redundant information.

3 Example Use Case

The department for National Highway Traffic Safety in the U.S. (NHTSA) wants to reduce the number of traffic crashes. For this purpose, they conduct recalls of unsafe vehicles and collect and analyze data on car crashes and problems with cars in a huge database since 1995. The data set contains structured information such as the car component affected. Customers filled this data field choosing the appropriate car component from a dropdown menu. Moreover, the NHTSA data set contains a free text field. The free text field describes the car crash or problem with the car. In Table 1 we show a small example data set containing information on the car component and a free text description.

Tab. 1: Example data set with structured (id, component) and unstructured information (description).

id	component	description
1	AIR BAG	AIR BAG FAILED DURING ACCIDENT (...)
2	AIR BAG	AIRBAG FAILED TWICE.
3	AIR BAG	AIR BAG LIGHT FAILED.
4	STEERING	VERY SENSITIVE STEERING AT HIGH VELOCITY (...)
5	STEERING	STEERING FAILED.
6	ENGINE	THE ENGINE SHUT OFF TWICE ON THAT DAY (...)
7	ENGINE	ALL ENGINE LIGHTS CAME ON (...)

An isolated analysis on structured data for example yields an ordered list of the most frequent car components involved in car crashes (cf. R_1 in Figure 1). An isolated analysis on the free text field also lists primarily car components (cf. R_2 in Figure 1). The results of Ghazizadeh et al. showed that half of the information in R_2 is not interesting to the analyst since it contains too much redundant information also contained in the structured field and i_{new} thus is comparably low ([GML14], see Section 2). They applied an isolated latent semantic analysis and hierarchical clustering approach to the NHTSA data set.

The hybrid approach tries to tackle this problem. It yields a list of frequently mentioned terms that are not deducable from the structured part of the dataset. R_1 and R_2 are oriented on car components, but R_3 mostly is oriented on issues. For example, in R_3 (cf. R_3 in Figure 1) the analyst finds new valuable information among the 5 highest ranked clusters (which will be discussed in more detail in Section 6.2): Many customers report problems in getting new secure car parts that the manufacturers need to change due to a recall. In isolated approaches the analyst misses this information since it is not present in R_1, and in R_2 it is ranked as place 175 only (see Section 6.2 for more details on the prototypical implementation yielding R_2). This information may be crucial in preventing car crashes. Customers, while waiting for the secure car parts, might decide to drive the car anyway. Furthermore, the information in R_3 can be used to improve and extend the categories available in structured data in a feedback loop. The analyst may decide to add the 'unavailability of car parts' to the future structured data values available to the customers who file a complaint in the NHTSA data base.

4 Hybrid Information Extraction Approach

The goal of our approach is to extract more information from structured and free text data in terms of a higher degree of new information as defined in Formula 1 in the introduction to this work. In Figure 2, we illustrate an isolated approach to information extraction and show the resulting table in Figure 3. In Figures 4 and 5, we show an example illustrating the hybrid information extraction method and R_3. Since we want to emphasize the difference between standard isolated approaches and our hybrid approach, we do not describe in this section preprocessing steps (such as tokenization) and vectorization, which both approaches have in common. We explain these steps in detail in the next section. Here, we focus on the steps special to the hybrid approach: (1) grouping and (2) removal.

In Figure 2 we illustrate an isolated approach. Here, free text fields are considered in isolation and all free texts are clustered. Finally, the name of the cluster in which a free text falls is added to the overall data set in the additional column 'cluster' as shown in Figure 3. The cluster name is based on the most frequent word in the cluster.

In Figure 4 we illustrate the first processing step of the hybrid approach, in which structured data is used to **group** free text fields. Here, the NHTSA data set is grouped by the structured data field on the car components into three groups (AIR BAG, STEERING and ENGINE) (step (1)). In Figure 5, we illustrate the next step, in which we **remove** all information that is already available in the structured data field on car components (step (2)). Only then, the free texts are **clustered** (step (3)). Finally we add a new column to the table which contains the name of the cluster, the result is shown in the last step in Figure 5. The isolated approach adds cluster information such as 'air bag' and 'steering'. This information is already available in the structured field 'car component' (see columns 'component' and 'cluster' in Figure 3). The hybrid approach results in clusters, such as 'light' and 'fail'. They represent new valuable information (see column 'cluster' and compare to column

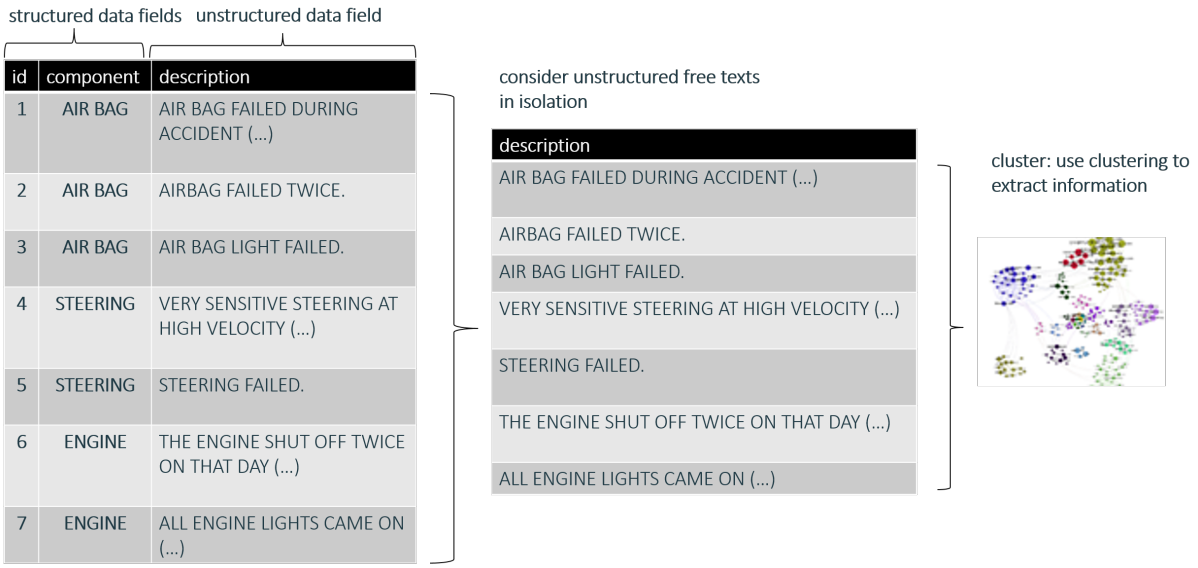

Fig. 2: Concrete example illustrating an isolated approach to information extraction from free text fields.

id	component	description	cluster
1	AIR BAG	AIR BAG FAILED DURING ACCIDENT (...)	air bag
2	AIR BAG	AIRBAG FAILED TWICE.	air bag
3	AIR BAG	AIR BAG LIGHT FAILED.	air bag
4	STEERING	VERY SENSITIVE STEERING AT HIGH VELOCITY (...)	steering
5	STEERING	STEERING FAILED.	steering
6	ENGINE	THE ENGINE SHUT OFF TWICE ON THAT DAY (...)	engine
7	ENGINE	ALL ENGINE LIGHTS CAME ON (...)	engine

Fig. 3: Concrete example illustrating the result of isolated information extraction from free text containing redundant information such as ’air bag’ and ’steering’.

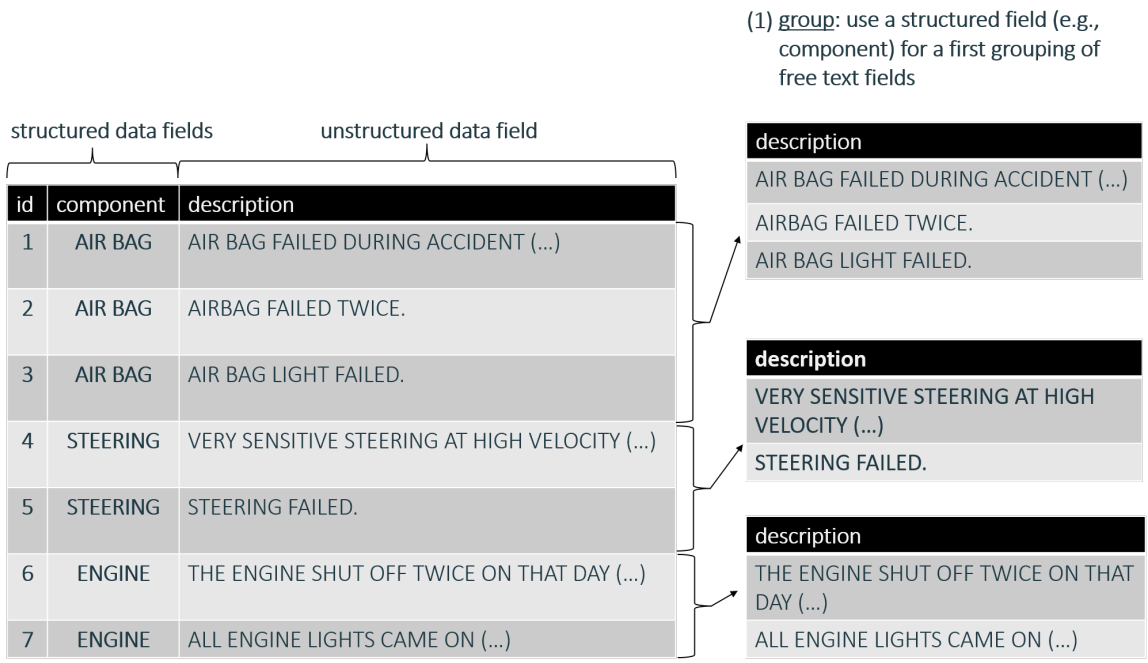

Fig. 4: Concrete example illustrating the distinguishing step 'group' of the hybrid approach to information extraction from free text fields.

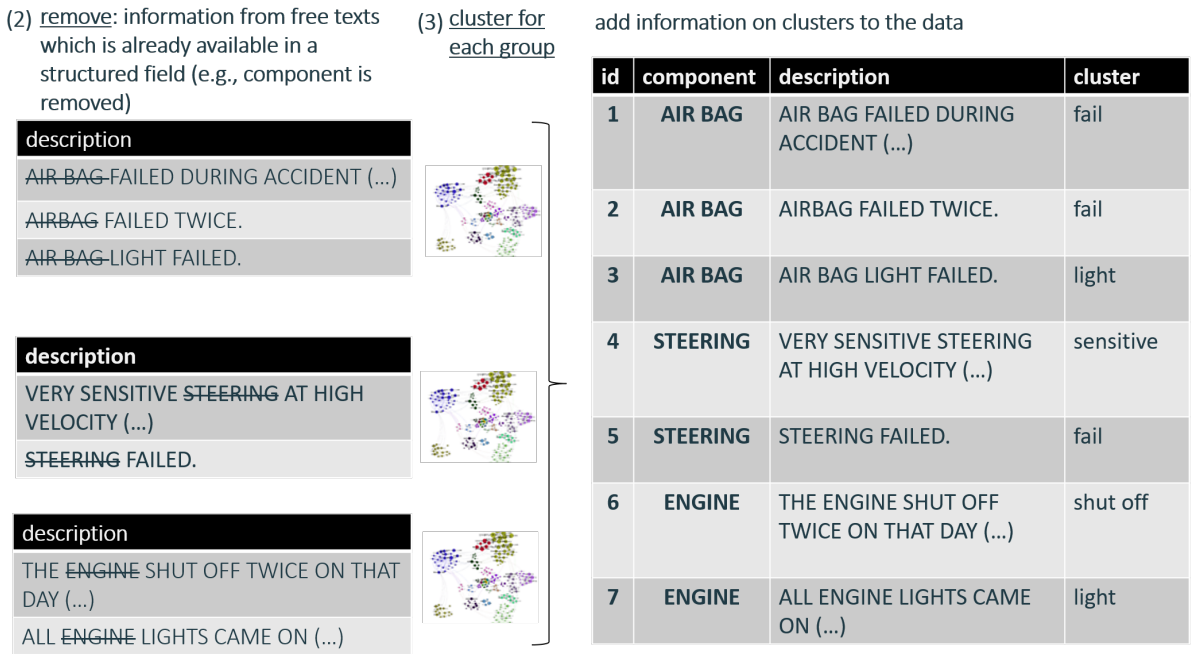

Fig. 5: Concrete example illustrating the distinguishing steps 'remove' and 'cluster for each group' of the hybrid approach to information extraction from free text fields and the result containing new valuable information such as 'fail' and 'light'.

'component' in Figure 5). We base our approach on three predominant characteristics of structured data sets with embedded free text fields, which we discuss in more detail in the remaining paragraphs of this section.

First of all, **free text fields store additional valuable information**. This was confirmed in many studies (see Section 2). Various methods, such as relation extraction, classification and clustering can extract valuable information stored in free text [ZM16]. We use clustering since it is suited best for our use cases. Moreover, Ghazizadeh et al. [GML14] also used a clustering approach, and we want to compare our results with their findings. However, the hybrid information extraction approach suggested in this paper is independent from the concrete information extraction method chosen. Also relation extraction and classification approaches may benefit from applying the concept to them. For example, a classifier which uses structured fields as well as unstructured free text fields in the feature generation, may benefit from removing information already present in structured fields from the free texts. The concrete effects on further information extraction methods need to be investigated in future work. Here, we focus on the validation of the core concept and employ a state-of-the-art clustering technique.

Second, in many data sets in industry and research, **we can group unstructured free text fields via information encoded in structured data fields**. For example, the NHTSA data set (see Section 6.1) may be divided into groups based on structured fields such as car component, year and car make. The hybrid approach uses this information for grouping. Thus, we do not end up extracting the same groups based on text mining free text fields. For a concrete example, see Figure 4, step (1).

Lastly, **if the same information can be extracted from either structured data or from free text fields of a data set, usually structured data is preferred**. In most research and industry data sets, the quality of structured data fields is estimated to be quite high. Pre-defined value ranges and quality control at the point of data entry lead to high quality structured data. However, the entry of texts is free and usually no pre-defined value ranges and quality control exist. Thus, free texts are oftentimes full of spelling mistakes, grammatical errors and abbreviations (compare e.g., [KM16] and [ZMZ16]). If an information is present in a structured field as well as in a free text field, we use the information from the structured field. Consequently, we do not want to extract this redundant information from the free text field with text mining. Thus, during preprocessing, we remove this information. E.g., in Figure 5 we remove the word 'steering' from all free texts in the respective group.

As we can see from Figure 5, in the hybrid approach, cluster names such as 'fail' or 'light' are added. After the grouping and removal step, these new cluster characteristics show up. Thus, compared to isolated approaches, our approach increases the amount of new information i_{new} available in the data set (see Section 6).

5 Design Decisions in the Implementation

The prototype is open source and can be retrieved from GitHub[6]. It is implemented in Python, since many Python programming libraries for natural language processing exist. The implementation is straightforward and helpful documentations are available (e.g., [BKL09] and [Pe14]). Furthermore, all tools and libraries chosen for the implementation of the prototype have industry-friendly licences. The prototypical implementation enables an easy integration of, e.g., new preprocessing methods, clustering algorithms and visualizations, as well as an easy adaptation to other languages. We designed the prototype that flexible so that both use cases with English and German free text fields may be covered easily. Other design decisions, such as on data types the prototype can read and preprocessing performed, are founded on the two use cases described in more detail in Section 6. In Figure 6, a schematic illustration of the prototype to our hybrid information extraction approach is shown. For the evaluation of the hybrid information extraction approach, we implemented a state-of-the-art clustering prototype as a baseline and a prototype for our hybrid approach. The two implementations are exactly the same, but for the two distinguishing processing steps 'group' and 'remove' (these two steps are bold-faced in Figure 6). In the following subsections we describe the processing steps from Figure 6 in more detail and state our implementation choices.

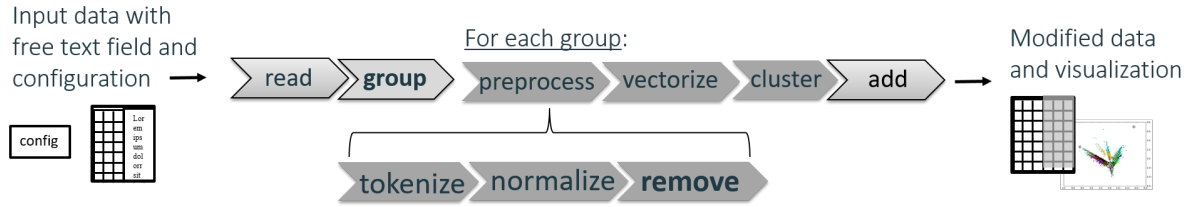

Fig. 6: Schematic illustration of the prototype. In the two boldfaced preprocessing steps 'group' and 'remove', structured data is used, which makes the approach hybrid.

For **reading** configurations, we used ConfigObj[7]. In the configuration, the user needs to state the column that contains the free texts and the column that contains the structured data that shall be used in the grouping and removal steps. If more than one structured categorical field is available and suitable, both may be applied to the 'removal' step. However, our use cases only require to select exactly one structured field for the 'grouping' step. Moreover, the processing steps can be freely defined by the user, or alternatively the default settings are used. With the default values, our prototype uses standard preprocessors, no synonyms in normalization, a tf-idf vectorizer and creates 12 clusters per group. The user may adapt these values if needed. ODBC data bases as well as CSV-formatted data sets may be read. Therefore, we use the library PyODBC[8] and a CSV-standard tool in Python[9]. NumPy[10] arrays represent the incoming and outgoing data.

[6] https://github.com/LinkMarco/PrototypeClustering
[7] `https://pypi.python.org/pypi/configobj/5.0.6`
[8] `https://pypi.python.org/pypi/pyodbc/4.0.3`
[9] `https://docs.python.org/3/library/csv.html`
[10] `http://www.numpy.org/`

We base the **grouping** step on Python standard tools and SQL SELECT statements which are invoked from Python. All following steps are subsequently executed for each group. The free texts are grouped based on the structured data column as defined by the user in the configuration. For a concrete example of this processing step, see step (1) in Figure 4.

The **preprocessing phase** is a central part in text mining, since here the features are given by the words of the text (see, e.g., [MRS08]). Table 2 shows the main steps.

The detailed settings are flexible and can be determined in the configuration. The main library we use for natural language processing is the Natural Language Toolkit (NLTK)[11].

Tab. 2: Small example illustrating the preprocessing steps of our prototype: tokenization, normalization, and the removal of stopwords and redundant information as determined via analysis of structured data.

Preprocessing step	Sample text
Before preprocessing	AIRBAGS FAILED DURING ACCIDENT, BUT CAR PARTS ARE NOT AVAILABLE.
Tokenize	[AIRBAGS] [FAILED] [DURING] [ACCIDENT] [,] [BUT] [CAR PARTS] [ARE] [NOT] [AVAILABLE] [.]
Normalize	[air bag] [fail] [during] [accident] [but] [car part] [are] [unavail]
Remove	[fail] [accident] [car part] [unavail]

For **tokenization**, the Whitespace Tokenizer[12] or the Penn Treebank Tokenizer[13] may be applied. In tokenization, the text is split into the smallest meaningful units such as words and compounds. We give an example in Table 2. Note that here the compound 'car parts', while being separated by a whitespace, was correctly selected as one token.

Then, we **normalize** all tokens. We provide plenty normalization methods in the prototypical implementation. These are optional and may be selected and adapted by the analyst for each use case, e.g., as described in Sections 6.2 and 6.3. In the normalization process, we may lowercase all texts. Spelling mistakes may be corrected using TextBlob[14]. Furthermore, contractions such as 'didn't' may be extracted. We use the multi-word expression tokenizer from NLTK[15] for extraction. Then, white spaces may be normalized (two or more whitespaces are reduced to one). Moreover, we may remove urls, mail adresses, telephone numbers, numbers, punctuation marks, currency signs and accents using tools from Textacy[16]. Finally, we may stem the tokens, i.e., we delete affixes with the goal of normalizing and thereby grouping the tokens. For stemming, we apply the Porter Stemmer[17] from NLTK. Additionally, different expressions which have the same meaning (=synonyms) may be consolidated. We

[11] http://www.nltk.org/

[12] http://www.nltk.org/_modules/nltk/tokenize/regexp.html#WhitespaceTokenizer

[13] http://www.nltk.org/_modules/nltk/tokenize/treebank.html#TreebankWordTokenizer

[14] https://pypi.python.org/pypi/textblob

[15] http://www.nltk.org/_modules/nltk/tokenize/mwe.html#MWETokenizer

[16] https://pypi.python.org/pypi/textacy

[17] http://www.nltk.org/_modules/nltk/stem/porter.html#PorterStemmer

implemented the synonym consolidation based on standard tools in Python. In the small example in Table 2, we lowercase all words. Then we normalize 'NOT AVAILABLE' to its synonym 'unavailable'. Finally we stem the tokens, which e.g., transforms 'unavail*able*' to 'unavail' and 'fail*ed*' to 'fail'.

We base the **removal step** on the Word List Corpus Reader[18] in NLTK. In this step, we remove stopwords. Stopwords are words that do not bear interesting information, but merely are present in the texts for grammatical reasons. In Table 2, *during, but* and *are* were identified as stopwords. Additionally we remove the information already present in the structured data column specified by the user. In our prototypical implementation, we add the string from the categorical structured field to the stopword list. Since these structured values usually are words in their base form, they match with the corresponding word occurences in the stemmed free text fields. Depending on the use case and data set, further resolutions of synonyms and abbreviations need to be added, which are also supported by the prototype. For a concrete example of the 'remove' processing step, see Figure 5 step (2). In the small example in Table 2, the word 'air bag' was additionally removed.

We use the machine learning library Scikit-learn[19] for **vectorizing and clustering** the free text fields. Vectorization means building a representation for each document that notes which words are present in the document and how these shall be weighted. The **vectorizer** can use two different weighting schemes: Either plain term frequencies (tf) or term frequencies times inverse document frequency (tf-idf). Tf-idf is a weighting scheme often used in information retrieval [MRS08], which leads to a proper baseline clustering prototype. Here, terms which are very frequent in the complete free text collection are downweighted and rare ones are upweighted. Thus, much redundant information is yet downweighted by means of the state-of-the-art weighting scheme. Thus, the state-of-the-art approach already extracts much new information and is a strong baseline. As we will show in Section 6, i_{new} still is higher for the hybrid information extraction approach than for that baseline. Several **clustering** algorithms are implemented in Scikit-learn. For it's popularity and robustness, we chose the k-means algorithm for the prototype. This algorithm is a hard partitioning clustering algorithm, which means that each free text may only be put into exactly one of the clusters built. K-means is implemented following Lloyd's algorithm [Ll06]. We use NumPy for array representations and calculations in Scikit-learn. Thus, vectorization and clustering is fast. The prototype is built so that it is possible to calculate and compare i_{new} in the evaluation of the core concept. While we employ a robust and state-of-the-art clustering algorithm in the prototype, the concept is independent of the implementation chosen. Since the clustering step in our prototype is based on the Scikit-learn library, it is easy to add other clustering algorithms if needed.

Finally, **a new column is added** to the original data set. For each data instance it contains the name of the cluster to which the data instance belongs. The cluster name is based on the most frequent word in the cluster. This processing step uses NumPy arrays and SQL.

[18] http://www.nltk.org/_modules/nltk/corpus/reader/wordlist.html
[19] http://scikit-learn.org/stable/

Visualizations are optional. We implemented them using Matplotlib[20]. The clusters as well as cluster quality metrics such as the silhouette coefficient may be visualized.

6 Evaluation of the Hybrid Information Extraction Approach

In the following subsections, we give details on two data sets from the product life cycle and explain how we apply the prototype to them. Furthermore, we compare the results of the hybrid approach with the results from isolated approaches. In Figure 1 these results are illustrated as R_1-R_3, where R_1 is the result of an isolated approach on structured data, R_2 is the result of an isolated approach on unstructured data, and R_3 is the result of our hybrid approach. For easy reference, we will denote the prototypes used in the experiments by R_1-R_3 respectively. In the following we define the three prototypes tested:

- R_1: The **isolated approach on structured data** is based on a simple SQL-query which we run on the databases. It is exemplified for the categorical structured data field 'component' in the NHTSA data set in the following SQL statement[21]. It simply groups the data for the structured data field 'component', counts the lines, and orders the result in descending order:

  ```
  SELECT component, count(*) FROM nhtsa_table
  GROUP BY component ORDER BY count(*) DESC;
  ```

- R_2: We base the implementation of the **isolated approach on unstructured data** on the prototype described in Section 5. In fact, it is exactly the same, but the grouping as well as the removal of information from free texts based on the structured data field are omitted. Standard stopwords such as *and, the, it* are still removed. This ensures that the effect of the steps special to the hybrid approach can be viewed in isolation.
- R_3: We described the implementation of the **hybrid information extraction prototype** in the previous section. It is adapted to the two use cases, i.e., as described below tailored configurations such as synonyms and preprocessors are defined.

By applying both R_2 and R_3 to the data sets, we can compare a state-of-the-art baseline clustering approach (R_2) with the very same approach plus the two distinguishing steps 'grouping' and 'removing'(R_3). Both R_1 and R_2 are oriented on the components, i.e., air bag, steering, power train, whereas R_3 mostly is oriented on issues, i.e., fail, noise, (unintended) acceleration, unavailable (car parts). In the presentation of our detailed evaluation results, we thus show for comparison the results R_1 representing component-based and R_3 representing issue-based results. Moreover, we report the difference between the three approaches in terms of the amount of new information i_{new} (as defined in Formula 1 in Section 1) and discuss the resulting clusters.

[20] http://matplotlib.org/

[21] Note: the 'component' column is named 'compdesc' in the original NHTSA data set.

6.1 Data Sets

To evaluate the prototype, we apply it to two data sets. The first one is a freely accessible data set from the National Highway Traffic Safety Administration (NHTSA) in the United States of America. Since the data set[22] as well as our prototype are freely accessible, all evaluation results with respect to the NHTSA data set are reproducible. The NHTSA complaint data set currently contains more than 1.3 million reports on incidents with cars.

The second data set used for illustration of our prototype comes from an industry partner in Germany. It contains 153k entries, comprises information on downtimes in a production line and contains German free text information. On that line, smaller, but complex parts of a car are manufactured. This data set allows us to see how the prototype may be applied to a use case in production. It contains structured information on the downtimes, such as error codes and the duration of a downtime in seconds. Furthermore, information on the reasons for downtimes and the actions that were taken to put the production line running again are noted in a free text field. The workers can fill the free text field via text entry into a tablet, directly on the shop floor. The text entries are in German and quite short (4.1 words per entry in average) and full of spelling mistakes and domain-specific abbreviations, which brings special challenges with respect to information extraction.

6.2 Data Analysis of the NHTSA Data Set

To apply the prototype resulting in R_3 to the NHTSA data set, at least the column that contains the structured field and the column which contains the free text field need to be given. We use the structured field with the affected car component in the grouping and removal steps. We stick with the default settings (see Section 5), but chose to add some synonyms and context synonyms. For example, all occurences of 'not' or 'failed' and 'deploy' with no more than 3 words in between are normalized to 'not deploy'. Thereby, different ways of expressing the same concept are normalized to one term. The added synonym consolidations help in building semantically reasonable clusters. In this use case they have no influence on the 'removal' step of our core method. From this hybrid prototype adapted to the use case, the isolated prototype for R_2 is deduced. Both approaches are the same but for the grouping and removing steps and generate the same number of clusters (12 per group). R_1 is computed as defined above.

In the first analysis, we focus on how an isolated approach on structured data only (resulting in R_1) and the hybrid approach (resulting in R_3) differ. In Figure 7, we show the five most frequent car components (R_1, left) and cluster terms (R_3, right). For easier readability and comprehensibility, we name clusters by the most frequent word in that cluster in the base form. From the structured data, we extract the information on the most frequent car components affected. E.g., the electrical system, air bags, power trains of automatic transmissions,

[22] NHTSA complaints data set: https://www-odi.nhtsa.dot.gov/downloads/

steering and power train. Using the hybrid approach (R_3), additional information can be extracted: We already know from structured data that problems with transmission are frequent. But we did not know that many complaints are about problems with noise, acceleration, unavailable (car parts) and problems where the car stalls. Accelerating the supply with car parts needed due to a recall (cluster 'unavailable') and investigating problems, where the car stalls (cluster 'stall') or accelerates (cluster 'acceleration') might help in preventing car crashes. The result R_3 completes the information already available. The user can use both in his analysis: structured as well as generated cluster information. In Table 3 we also note both, information from R_3 and R_1 (the latter is noted in brackets following the free text samples).

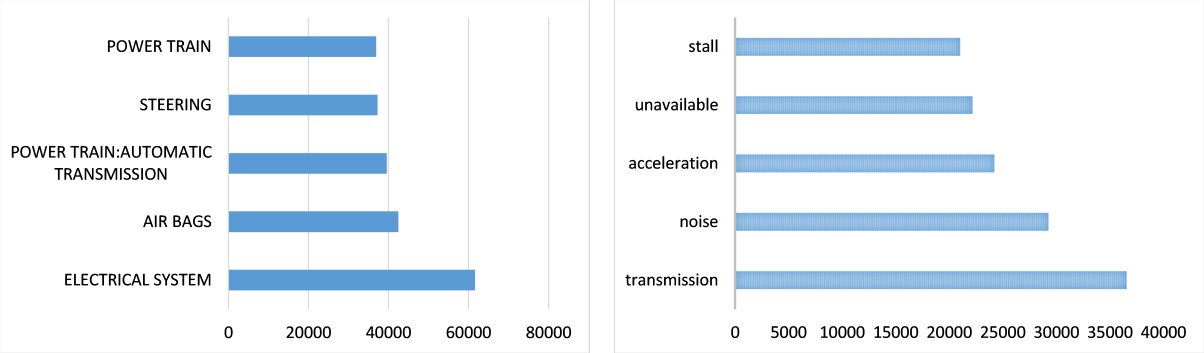

Fig. 7: Most frequent car problems based on an isolated analysis of structured data only (R_1, left) and on the hybrid analysis (R_3, right), all based on the NHTSA data source.

Tab. 3: Concrete examples of NHTSA complaints for most frequent clusters together with structured information as noted in the categorical field 'component' given in parenthesis

Cluster	Sample complaints
stall	VEHICLE STALLED DUE TO AN ELECTRICAL PROBLEM.('component': 'electrical system'); ENGINE STALLS WHEN APPROACHING A STOP. ('component': 'power train')
unavailable	THE PART TO DO THE REPAIR WAS UNAVAILABLE. (...) ('component': 'air bag'); (...) PARTS FOR RECALL IS NOT AVAILABLE SINCE REQUESTING (4) WEEKS AGO. (..) ('component': 'child seat')
acceleration	VEHICLE ACCELERATED BY ITSELF (...) ('component': 'vehicle speed control'); THE VEHICLE IS NOT ACCELERATING PROPERLY. (...) ('component': 'power train:automatic transmission')
noise	IT MAKES A LOUD NOISE AND NO 1 KNOW WHAT IT IS ('component': 'electrical system');WHINING NOISE WHEN TURNING STEERING WHEEL. (...) ('component': 'steering')
transmission	TRANSMISSION FAILURE AT 105,000 MILES (...)('component': 'power train:automatic transmission')

In a last experiment with the NHTSA data, we want to check how R_1-R_3 differ in terms of i_{new} (see Formula 1 in Section 1). In R_1 no new information which goes beyond the structured information may be gathered, thus $i_{new} = 0$. We compare the degree of new information within the 100 biggest clusters in R_2 and R_3. We illustrate the difference between

Tab. 4: Comparison of the degree of new information i_{new} for the three approaches R_1-R_3 on the NHTSA data set

Result set	i_{new}
R_1 (only structured information)	0
R_2 (baseline)	0.55
R_3 (hybrid approach)	0.98

R_1-R_3 with respect to this measure in Table 4. Here we see that R_3 contains significantly more new information than R_2. Also see the results already described in Section 3. The new information might be crucial for manufacturers, e.g., to get better customer satisfaction. Thus, we were able to confirm and improve the results of Ghazizadeh et al. [GML14], who report half of the cluster names to correspond to vehicle components, which corresponds to an i_{new} value of 0.5.

6.3 Data Analysis of the Industry Data Set

For a second evaluation, we apply the prototypes to an industry data set with a German free text field. It contains information on causes and actions related to machine downtimes. We use a structured field with an error code description which indicates the group of errors of a downtime on the production line for grouping. The possible choices for filling the structured data field do not cover all types of errors and reasons for downtimes that may occur in reality. More choices are indicated in a free text field and may be deduced by our hybrid information extraction method, only.

To apply our prototype resulting in R_3 to the industry data set, details in preprocessing need to be changed in the configuration file due to German text: a German standard stopword list (from NLTK, see Section 5) and a German stemmer[23] need to be specified. We normalize some spelling mistakes and verbalizations and add a few synonyms and context synonyms. For example, if the german words 'strom' and 'leerlaufstrom' (both terms are on power/electricity) are mentioned near 'hoch' (english 'high') the main word is substituted by 'strom_hoch'(in english 'power_high'). Some of the added synonyms help ensure a good recall of the 'removal' step. Also, encoded umlauts such as 'ae' and 'ue' are normalized to 'ä' and 'ü' respectively. After adapting the hybrid prototype to the use case, R_2 is deduced from it. Both approaches yield the same number of clusters and only differ in terms of the two distinguishing steps 'group' and 'remove'. K in k-means is set as described above for the NHTSA data. For R_1, the data is grouped by error code description with a SQL statement similar to the one described in the introduction of this section. For reasons of confidentiality, we use high-level abstractions of the German definitions. We furthermore translated them to English for the examples given in this work.

[23] e.g., `http://www.nltk.org/_modules/nltk/stem/snowball.html#GermanStemmer`

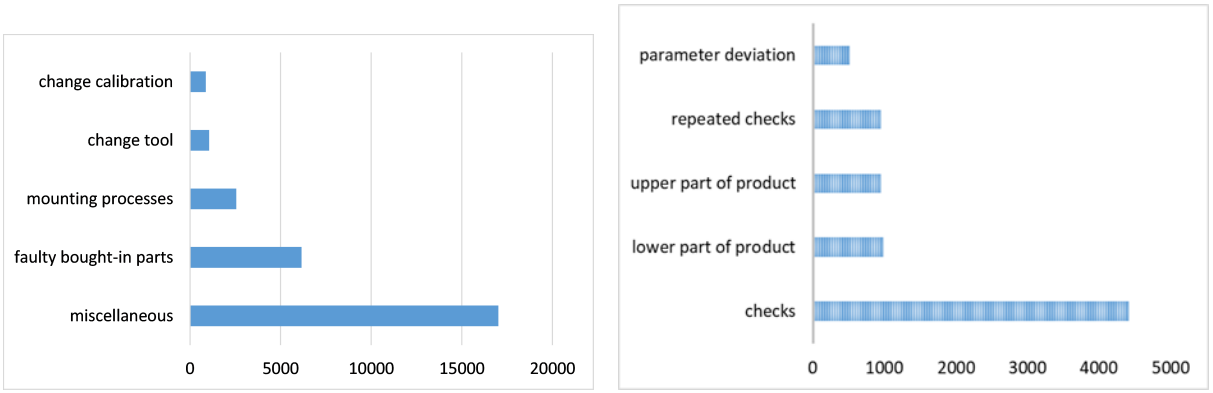

Fig. 8: Most frequent reasons for downtimes based on the structured data field containing an error code (R_1, left) and based on the hybrid approach (R_3, right), all based on the industry partner data source.

Finally, we compare the results of an isolated approach on structured data only (R_1) with the results of the hybrid approach (R_3). Using an isolated approach on structured data, the reasons for downtimes may be analyzed using the structured field with an error code and a table that defines these error codes. Following the hybrid approach, many fine-grained clusters are gained that represent new information. The results are illustrated in Figure 8. We see that the structured information on errors on the production line are very coarse-grained: problems due to faulty bought-in parts, mounting processes in general, change of tools and change of calibrations often lead to downtimes. The most frequent error code hints at 'miscellaneous' problems. This group is very big since many reasons for downtimes are not reflected in the given structured data values. Therefore the workers on the shop floor oftentimes chose the structured value 'miscellaneous' instead.

This is not helpful for the workers on the shop floor, who want to prevent or fix problems with downtimes of a production line. In contrast, the clusters resulting by our hybrid approach (R_3) are much more fine-grained. For example, in the big structured group 'miscellaneous', clusters such as 'problems with component parts' and 'problems with out-of-commission machines' were found. These give more detailed information to workers on the shop floor and to managers of the production line. From R_3, more detailed and new information on reasons for downtimes may be extracted: From biggest to smaller clusters, detailed information on problems with checks, parts of the products which are produced, repeated checks and parameter deviations that lead to downtimes in the production line are reflected in the clusters. If this data was prepared for the shop floor workers, it might help in solving new downtimes of the production line faster. In a feedback loop, new reasons for downtimes in the production line may be added to the list of error codes in order to strengthen the significance of the structured fields.

We moreover compare the degree of new information in Table 5 with the same method as described for the NHTSA data set in the previous section (see Formula 1 in Section 1). In this case the baseline is even stronger, due to weaker structured information available in

Tab. 5: Comparison of the degree of new information i_{new} for the three approaches R_1-R_3 and the industry data set

Result set	i_{new}
R_1 (only structured information)	0
R_2 (baseline)	0.78
R_3 (hybrid approach)	0.99

the industry data. Still, R_3 has a 0.21 higher i_{new} value than R_2. The grouping step helps to reduce the number and size of big 'miscellaneous' clusters. Special attention needs to be payed to synonyms, spelling mistakes and abbreviations. If not adressed properly, these issues may lead to less exploitation of the benefits of the removing step. We discussed the different results with domain experts and found that the additional information contained in R_3 is relevant to the task of optimizing the process. Also, from the point of view of domain experts, the hybrid information extraction method has future potential: Before a new shift on the production line begins, a summary of current insights may be presented to the shop floor workers. Moreover, new staff could be assisted by a presentation of aggregated insights.

7 Conclusion and Future Work

We suggested a hybrid approach to the extraction of information from free text fields which yields more new information i_{new} from data with structured data fields and unstructured free text fields. The approach is based on natural language processing and k-means clustering and improves the results of two baseline isolated approaches by employing analytical results from structured data within the text analysis process. First, we group data based on a structured data field. Then, we preprocess data, while also redundant information, as determined via a structured data field, is removed. Then, data is clustered and a new column containing the cluster name is added to the data set. In this paper, we describe the concept and implementation of our approach for hybrid information extraction and discuss relevant design considerations. We based the prototype for the two baselines as well as the hybrid information extraction approach on free, open-source tools. The prototype is freely available on GitHub[24]. The prototype for R_2 can be deviated from it and R_1 can be gathered by means of a SQL-query as shown in Section 6. Finally we evaluated our approach with two example data sets with German and English free text fields. While the data set from production is confidential, the NHTSA data set may be downloaded[25] and thus the results presented with respect to this use case are reproducible. We compared our approach to baseline isolated approaches on structured or unstructured data. We showed that isolated approaches to free text yield much redundant information already available in structured

[24] https://github.com/LinkMarco/PrototypeClustering
[25] https://www-odi.nhtsa.dot.gov/downloads/

data. The hybrid approach impedes this and yields more new information. For the two use cases, the degree of new information in R_3 is significantly higher than in R_2.

In future work, we integrate our hybrid information extraction approach into a framework of methods for measuring and improving data quality of product lifecycle data. Then, we will apply the concept to further information extraction approaches. In future work, an efficient handling of big data may be enabled by transferring the prototype into text databases (e.g., [KLA15]). Moreover, we will address issues we encountered in analyzing production data such as synonyms, spelling mistakes and abbreviations. Furthermore, we will employ additional evaluation metrics, e.g., based on entropy.

Acknowledgements

The authors would like to thank the German Research Foundation (DFG) for financial support of this project as part of the Graduate School of Excellence advanced Manufacturing Engineering (GSaME) at the University of Stuttgart. Moreover, we thank Marco Link for crucial implementation work.

References

[BKL09] Bird, Steven; Klein, Ewan; Loper, Edward: Natural Language Processing with Python. O'Reilly Media, 2009.

[Br08] Brooks, Benjamin: Shifting the focus of strategic occupational injury prevention: Mining free-text, workers compensation claims data. Safety Science, 46(1):1–21, 2008.

[Ce15] Ce Zhang: DeepDive: A Data Management System for Automatic Knowledge Base Construction. PhD thesis, University of Wisconsin-Madison, 2015.

[CH14] Carter, B.; Hofmann, M.: An analysis into using unstructured non-expert text in the illicit drug domain. In: 2014 IEEE International Advance Computing Conference (IACC). pp. 651–657, 2014.

[CRB11] Chougule, Rahul; Rajpathak, Dnyanesh; Bandyopadhyay, Pulak: An integrated framework for effective service and repair in the automotive domain: An application of association mining and case-based-reasoning. Computers in Industry, 62(7):742–754, 2011.

[FKS06] Forman, George; Kirshenbaum, Evan; Suermondt, Jaap: Pragmatic Text Mining: Minimizing Human Effort to Quantify Many Issues in Call Logs. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '06, ACM, New York, NY, USA, pp. 852–861, 2006.

[Ga05] Gamon, Michael; Aue, Anthony; Corston-Oliver, Simon; Ringger, Eric: Pulse: Mining Customer Opinions from Free Text. In: Proceedings of the 6th International Conference on Advances in Intelligent Data Analysis. IDA'05, Springer-Verlag, Berlin, Heidelberg, pp. 121–132, 2005.

[GML14] Ghazizadeh, Mahtab; McDonald, Anthony D.; Lee, John D.: Text Mining to Decipher Free-Response Consumer Complaints: Insights From the NHTSA Vehicle Owner's Complaint Database. Human Factors The Journal of the Human Factors and Ergonomics Society, (56, 6):1189–1203, 2014.

[GSB14] Gubanov, M.; Stonebraker, M.; Bruckner, D., eds. Text and structured data fusion in data tamer at scale: 2014 IEEE 30th International Conference on Data Engineering, 2014.

[HW96] Hogan, W. R.; Wagner, M. M.: Free-text fields change the meaning of coded data. Proceedings of the AMIA Annual Fall Symposium, pp. 517–521, 1996.

[KLA15] Kilias, Torsten; Löser, Alexander; Andritsos, Periklis: INDREX: In-database relation extraction. Information Systems, 53:124–144, 2015.

[KM16] Kassner, Laura; Mitschang, Bernhard: Exploring Text Classification for Messy Data: An Industry Use Case for Domain-Specific Analytics. In: Advances in Database Technology - EDBT 2016, 19th International Conference on Extending Database Technology, Proceedings. OpenProceedings.org, pp. 491–502, 2016.

[Ll06] Lloyd, S.: Least Squares Quantization in PCM. IEEE Trans. Inf. Theor., 28(2):129–137, 2006.

[MRS08] Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich: Introduction to information retrieval. Cambridge University Press, New York, 2008.

[Pe14] Perkins, Jacob: Python 3 text processing with NLTK 3 cookbook: Over 80 practical recipes on natural language processing techniques using Python's NLTK 3.0. Packt Pub., Birmingham, UK, 2014.

[SBP06] Silva, E. F. A.; Barros, F. A.; Prudencio, R. B. C., eds. A Hybrid Machine Learning Approach for Information Extraction: 2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06), 2006.

[Ta00] Tan, Pang-Ning; Blau, Hannah; Harp, Steve; Goldman, Robert: Textual Data Mining of Service Center Call Records. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '00, ACM, New York, NY, USA, pp. 417–423, 2000.

[XZZ08] Xiao, Ji-Yi; Zhu, Dao-Hui; Zou, La-Mei, eds. A hybrid approach for web information extraction: 2008 International Conference on Machine Learning and Cybernetics, volume 3, 2008.

[ZM16] Zhai, ChengXiang; Massung, Sean: Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining. ACM, New York, NY, USA, 2016.

[ZMZ16] Zhang, Yuhao; Mao, Wenji; Zeng, Daniel: A Non-Parametric Topic Model for Short Texts Incorporating Word Coherence Knowledge. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. CIKM '16, ACM, New York, NY, USA, pp. 2017–2020, 2016.