HAW
HAMBURG

Bachelorarbeit

Gerriet Hinrichs

Data analyses and preparation for machine learning based order prediction

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Gerriet Hinrichs

Data analyses and preparation for machine learning based order prediction

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 1.7.2019

Gerriet Hinrichs

Title of Thesis

Data analyses and preparation for machine learning based order prediction

Keywords

Data analyses, data preparation, machine learning, order prediction

Abstract

The thesis discusses label generation for order positions based on free-text identifiers for a later predictive analysis with the goal to optimize a business process. Working with real-world data, analyses show the low data quality and the need to generate appropriate labels. With a generic data processing architecture, an iterative approach is taken to create a label approximation. It is reasoned that data quality is important and that poor quality might prevent useful data analyses.

Gerriet Hinrichs

Thema der Arbeit

Datenanalyse und Aufbereitung für die Bestellungsvorhersage mittels maschinellem Lernen

Stichworte

Datenanalyse, Datenaufbereitung, maschinelles Lernen, Bestellungsvorhersage

Kurzzusammenfassung

Die Arbeit diskutiert Label Generierung für Bestellpositionen auf Basis von Freitextbezeichnungen für eine spätere prädiktive Analyse mit dem Ziel einen Geschäftsprozess zu optimieren. Analysen der verwendeten Daten aus der Wirtschaft zeigen die geringe Datenqualität und die Notwendigkeit vernünftige Labels zu erzeugen. Mit einer generischen Datenverarbeitungsarchitektur wird in einer iterativen Herangehensweise eine Label-Approximation erzeugt. Es wird geschlussfolgert, dass Datenqualität wichtig ist und dass schlechte Qualität sinnvolle Datenanalyse verhindern kann.

Contents

List of Figures

Acronyms

KDD Knowledge Discovery in Databases.

LSTM Long Short-Term Memory.

NLTK Natural Language Processing Toolkit.

1 Introduction

Within modern big-data systems, predictive analyses are often used to improve business processes. The need for such optimizations can also be found in modern shipping industry. Due to the growing competition and the resulting need to operate as cost efficient as possible, intelligent supply management is one of the topics for business optimization.

While intelligent supply management is done for a few years within other industry, precise business intelligence about the involved processes is rare in shipping industry. With more data becoming available, issues with data quality often occur as data is most of the time collected without having data analysis in mind.

In our specific case, order positions are not easy to identify for an intelligent system. As such identification is crucial for supply process analysis, a way of order position identification is needed.

This thesis is separated into five chapters. Within chapter 2, the problem itself is described. We're also having a closer look at the basics of data analysis and are discussing comparable work. The goal for this thesis is then defined.

Chapter 3 deals with the used data. We're working with a data set provided by Hanseaticsoft GmbH, a software provider for shipping companies. Available data has to be categorized, relevant data has to be selected and there might be the need for additional information from other sources. It is then explained why order position identification is not directly possible.

Within chapter 4 we're working on solving the identification issue. Starting with a general software architecture, we're using an iterative approach to properly identify order positions. We're then evaluating our final result.

The summary in chapter 5 finalizes this thesis. Our approach and the result is summarized. We're reasoning that bad data quality can be an issues for or even prevent useful data analyses.

2 Problem Analysis

For machine learning based order prediction, data analysis and preparation is required. We're aiming to optimize a business process involving orders but data quality issues have to be solved first.

2.1 Business Context

Ongoing digitalization and process optimization is a crucial part of modern shipping industry [Lind et al., 2015].

One of these processes deals with requisitions: Supply requests from vessels. As vessels often request supplies late, in some cases only a few hours before the vessel reaches the port the supplies are ordered to, contacting suppliers and organizing the delivery can be costly.

Because late supply requests result in unnecessary costs that can be prevented by better resource planning, knowing supply demands early improves this specific business process. This project aims to create a forecasting model for such supply requests based on history data. Such a forecasting model would allow the office to buy the required supplies and have them delivered at favorable conditions.

2.1.1 Supply Process

The general supply process is structured into four parts: The requisition, order, delivery and invoice processes. During these steps, a supply request might change.

The requisition process deals with the actual supply request of a specific vessel. A supply order is created by the vessel and then reviewed by the office. The original supply request might be changed during this process.

The order process starts once the requisition has been reviewed. It deals with ordering the required supplies from a supplier. During this step supply items might be replaced by a suitable substitute, further changing the original supply request.

The delivery process then deals with transport of any ordered supplies to the vessel. Mostly consisting of transport organization, this part also deals with missing or broken items.

The invoice process finalizes the whole supply process and covers billing. During this step, the order itself is no longer modified, but billed positions might be different to the actually delivered items.

2.2 Machine Learning

Machine learning is a part of artificial intelligence and describes machine and algorithm supported generation of knowledge based on data. It also describes solving data based problems using this generated knowledge. Within this thesis, machine learning is used as group of algorithms used for knowledge discovery and generation. If focuses on data processing, one of the first steps of knowledge discovery.

Anders [2018] stated that machine learning, compared to other statistic methods, works well for large amounts of data or data with a very complex structure. But he pointed out two major problems with such methods: The non comprehensible results of some algorithms and the replacement of human intuition and background knowledge.

Non comprehensible results of some machine learning algorithms can be a risk for critical production systems, especially if safety or quality guarantees are required.

The replacement of human intuition is another issue for the use of machine learning algorithms. Although observations suggest that properly trained machine learning models gives better results than human intuition, neglecting, e.g., business experience is subjectively wrong for many people.

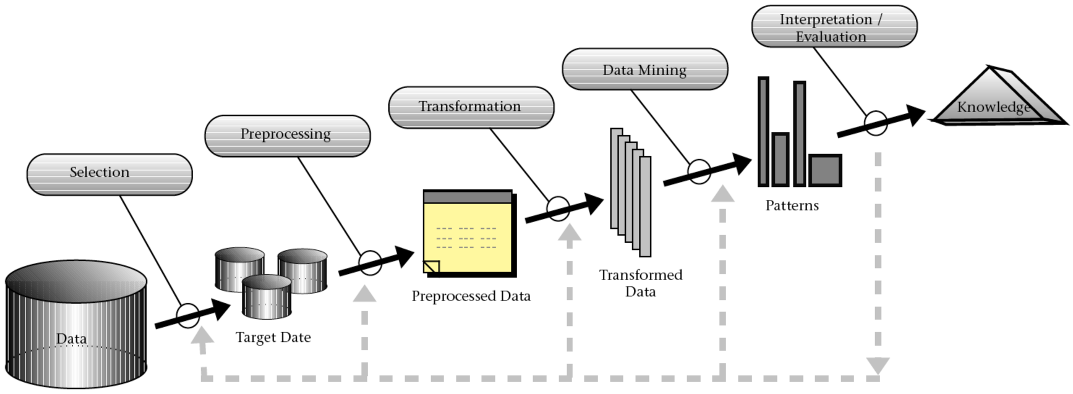

Figure 2.1: KDD process as described by Fayyad et al. [1996]

2.3 Development of Intelligent Systems

Shipping business changed rapidly within the last decades. During the ongoing process of digitalization, more data is collected and many more data dimensions become available. The main reason for this is the use of new technology and infrastructure in seafaring. Many vessel components (e.g., the main engine) are monitored with way more precision and shipping in general becomes more connected. Some reasons for this are the growing international competition or new laws like the EU regulation regarding CO_2 emissions [Council of the European Union, 2016]. On the other hand, cloud storage became affordable for small companies so all that new data can be stored cheaply.

However, huge amounts of data or complex data structures with many dimensions are difficult to process and analyze. Due to that, processes were developed for building intelligent systems that allow handling such data. One of those is the general process of *Knowledge Discovery in Databases (KDD)* as described by Fayyad et al. [1996].

The KDD as shown in figure 2.1 consists of five steps. Starting with raw data, smaller data sets for training and verification are selected in the first step. These data sets are then preprocessed during the second step to reconstruct missing data, remove faulty data entries or extract additional information using simple statistic methods. During the third step, data is transformed into a data structure that can be used for data mining. The fourth step contains the actual data mining using various machine learning algorithms.

Evaluating the mining result is part of the last step. This step contains verification of trained models using the verification data sets and evaluating the gained knowledge based on the initial KDD goals.

The problem with KDD is that it was build for smaller and manually maintained databases. Therefore it is not always possible to apply it to modern knowledge discovery processes. However, the basic procedure is still valid and many knowledge discovery processes are based on KDD. That also applies to the process used within this thesis.

A crucial part of the KDD process is the data selection step. It consists of two parts, the extraction of all usable data from various sources and the separation of the collected data into multiple training and verification data sets. Within this step available data has to be analyzed, especially in terms of implausible and missing data.

It is followed by the preprocessing step that mainly deals with feature engineering. Synthetic generation of values for missing data can also be part of this KDD step.

The work of this thesis belongs to these first two KDD steps.

2.4 Order Prediction

The targeted business process optimization as described in chapter 2.1 requires an order prediction model. To create such a prediction model, data classification is needed.

2.4.1 Prediction Approaches

There are two general approaches to create such a prediction model. The first way is to look at situation descriptions for data classification. This has been done by, e.g., Bogina et al. [2016]. They reduce log series data to situation descriptions using trend analyses.

Building the prediction model based on time series is the other general approach. This has been done by, e.g., Anders [2018] who used Long Short-Term Memory (LSTM) networks.

2.4.2 Foreground & Background Knowledge

Foreground knowledge within our prediction scenario contains past supply orders and their specific order books.

The belonging background knowledge is way more complex. It can be reduced to either a complex situation description or various series of specific events for different context aspects. This allows the usage of both general prediction approaches described in chapter 2.4.1.

2.4.3 Problematic Foreground Noise

The biggest issue with the used data set is noise within the foreground knowledge due to missing explicit labels on all items. For classification to work, proper labels have to be extracted from the data first. To achieve this, text analysis and text mining algorithms are used. This gives a usable item label approximation based on educated guesses.

2.4.4 Text Analysis & Text Mining

Text mining on short item names is not a common research problem. However, Twitter text analysis is and there is not a big difference in the way of doing it. These Tweets are rather short and mostly contain only one or two sentences. The main difference is that Tweets have specific context that can be used to further analyze their contents.

A review of different Twitter sentiment analysis techniques done by Bhuta et al. [2014] also includes text mining and text processing. A lexicon based approach and a Naïve Bayes one are the two related analysis techniques.

The lexicon based approach works with word counting. Positive or negative meaning of these words is annotated by the lexicon. In addition, opinion words and topics are extracted. The main disadvantage of this approach is that the word context is ignored.

The Naïve Bayes approach works with a probabilistic classifier. Using a decision system, the best matching text hypothesis is selected.

Twitter text preprocessing done by Singh & Kumari [2016] focuses on text normalization. Starting with punctuation, stop words and attached word splitting, multiple sequential preprocessing steps are performed. Then a spell checking algorithm is used and slang words are replaced by their counterparts. After folksonomy separation, the text is normalized.

2.5 Goal

The work of this thesis belongs to the steps one and two of the KDD process as described in chapter 2.3. Training data has to be selected and only a subset of the data dimensions is relevant for the actual problem. The focus is on labeling the requested items as this information is required for later KDD steps.

Based on the supply orders, we're building up an order book with hierarchically classified items. The groups or categories within this order book are created by text mining and reflect textual similarity of items.

2.5.1 Ground Truth

For the order prediction, we have to take a closer look at the actual ground truth for orders. As supply requests can be altered during the supply process as described in chapter 2.1.1, we have a possible difference between the requested items and quantities, and the actually delivered items and quantities. We also have information on valid item substitutes.

As this information is quite important to get good results during order prediction training, we're ignoring it for the item name preparation and keeping it in mind for our next steps towards the actual order prediction.

2.5.2 High Dimensional Feature Vector

As the used data provides many information related to the supply process, the high dimensional feature vector has to be analyzed. We need to find out what information is contained and whether this information needs further preparation.

The most interesting aspect of the feature vector is the difference between the originally supply request sent from the vessel and the actually delivered items. As these delivered items are closely related to the requisition items, the same issues with data quality arise. These delivery items have to be labeled as well.

2.5.3 Understanding Orders

Boiling down the tasks ahead, we need to get an understanding of the supply orders. We need to identify the items requested by the vessel and need to get the requested and actually delivered items and their volumes. With this information available, we're able to work on a supply order prediction model.

3 Data Analysis

The data set for this thesis is provided by Hanseaticsoft GmbH. It contains, starting with single requisition items, a few hundred thousand records with almost four thousand data dimensions. Technical information about the vessel, details about crew members and current cargo, as well as various information regarding the vessel's current schedule and consumption is included.

3.1 Data Categories

The data can be categorized into several basic groups describing different aspects of a supply order and its context.

3.1.1 Supply Process

The first category covers the supply process as described in chapter 2.1.1 and contains all of its required entities. This includes the supply orders from vessels, the belonging item positions, inquiries and orders sent to suppliers, all delivery information, and billing data. Data in this category covers all required information for organizing vessel supplies.

3.1.2 Vessel & Maintenance

A detailed vessel description together with its maintenance state forms the second data category. Entities within this category describe the vessel's type and its components down to specific parts of equipment. For each of these parts, maintenance and inspection data is available.

3.1.3 Schedule & Charter

Another important aspect is the vessel's current journey. This category contains the planned and actually traveled route, the vessel's cargo and running hours of various components together with their consumptions.

Furthermore, charterer information, together with associated billing data, is contained within this category, as the charterer has direct influence on several aspects of the vessel's journey.

3.1.4 Crew

Crew related information forms the last data category. The seamen on board and all their personal data, consisting of their nation, social data, rank together with their payroll, is contained. This information is directly related to the vessel's schedule, as the crew changes during long journeys.

3.2 Relevant Entities

While the data set in general has a complex structure, in the context of this thesis only a few key parts are important. Most of the available information describes the overall context of supply orders and their belonging vessels, but we're focusing on the actual supply requests for now.

3.2.1 Requisition Item

Requisition items describe supply request positions requested from a vessel. They are grouped by a requisition describing a single equipment or supply request.

The actual item is either identified by a reference to a catalog item or a free-text name and item number. In addition to the requested quantity together with a unit code, some metadata is available. Starting with the quantity approved by the office, a remark containing additional information about the item and a flag if the item is considered an unexpected expense is available.

A label for these items is not directly available, but the referenced catalog item could be used if present. For any other case, the free-text item name has to be analyzed.

3.2.2 Catalog Item

Catalog items identify actual items provided by suppliers in a standardized way. They are often imported based on a supplier catalog or created by the office for standard equipment and supplies.

The catalog item contains the item's name, unit code and item number. In addition, catalog items are assigned to a specific catalog and a, possibly nested, logical category.

3.2.3 Special Position Metadata

Within the actual order process, additional billing only positions are required. These position describe additional costs related to the specific order but don't provide actually delivered items themselves. Special positions can be packaging or freight costs, or describe tax fees. As these special positions have to be handled differently for billing and budgeting, metadata entities for them exist.

The special position metadata entity simply contains the name of the special position type and various information regarding billing and budgeting.

3.3 Additional Background Knowledge

In addition to the data provided by Hanseaticsoft GmbH, other information is needed for more sophisticated text processing. For this thesis, knowledge regarding language processing is enough, but the data set in general can be expanded with public available nautical, commercial and weather data easily.

3.3.1 Part of Speech & Stemming

As more sophisticated text processing was needed while analyzing the free-text item names, the Natural Language Processing Toolkit (NLTK) [Loper & Bird, 2002] was used. It contains two parts that are important for later analyses: An annotation system for part of speech information and stemming algorithms.

The part of speech tagging provided by the NLTK gives important information for more sophisticated text structure analyses.

Reducing different word forms to their word stem using the provided stemming algorithms allowed to further enhance later item categorization.

3.4 Insufficient Labels

We're working with a subset containing 98 598 requisition items. First analyses showed that only 12 570 of these items (12.75%) are ordered from catalogs. Any other item is only specified by a free-text string and item number.

These free-text items don't share a common naming scheme. In fact, typing errors, incorrect spelling, and inconsistent text formats can be found. Because of that, free-text names can't be used as proper item labels.

3.4.1 Inconsistent Text Formats

Looking at the free-text item names, the most obvious problem is the inconsistent text format as most items don't share a noticeable structure. These inconsistencies can be grouped into the following aspects.

Different capitalization is the most easy to handle one. Some items are written in all upper or lower case (e.g., “atomizer” vs. “ATOMIZER”), some have mixed capitalization following no convention (e.g., “Disinfectant & Antiseptic Hand Liquid”, “Hand cleaning wipes” or “Naloxone Hydrochloride 0.4mg in 1ml/3 ampoule”).

Varying separators are commonly found. This is most notable on the simple item “o-ring” that is further identified by a specific item number. In some cases no separator exists at all (“oring”) or a simple space is used (“o ring”).

Those variations also occur within logical segments of the name. For most items, the separate item number field is used, but the item name itself is also used for specifying the exact item (e.g., “THERMOMETER 0... +150°C R1/406/L130” or “Sensor Probe for T2000-TFC-02”).

Word order varies as well. Wording variations (e.g., “stainless steel” vs. “steel, stainless”) can be found. They sometimes introduce a different logical structure (e.g., “Disposable Gloves” vs. “Plastic Gloves (disposable)”).

3.4.2 Typing Errors & Spelling

As the free-text names are manually entered, common typing errors can be found. Most notable two are swapped order of adjoined letters (e.g., “Omhmeter”), other typing errors are rarely found.

In addition to simple typing errors, there are also issues with spelling itself (e.g., “Batteries”, “Battaries” or “Battreies”). This issue occurs the most, compared to the other typing and spelling errors.

3.4.3 Pieces Unit Code

Another interesting phenomenon is the pieces unit code. While there is a unit code field on the requisition item, many items simple use piece as unit. In many of these cases, the actual unit code is contained within the item’s name (e.g., “CASTOR OIL INDUSTRIAL 17KGS”).

This introduces another inconsistency between items, as some use the correct unit code, some actually have pieces as correct unit and others have their unit contained in the item name. Detecting and handling these cases properly adds more complexity to the item handling.

4 Data Processing

As the requisition items don't provide suitable labels themselves, we need to approximate them. This requires text processing of the free-text names. We're taking an iterative approach, evaluating the results of each iteration and then adjusting our text processing and label generation for the next iteration.

4.1 General Architecture

The software architecture for item labeling, as shown in figure 4.1, consists of two major parts: Data warehouse and processing pipeline. While the data warehouse part simply provides raw input data extracted from various sources [Inmon, 1996], the processing pipeline is a bit more complex. It consists of multiple processing steps dealing with data selection, data analyses and knowledge generation. These processing steps are largely based on each other but can be reordered or replaced to improve the overall result. Even within each processing step, the chosen algorithm can be changed without invalidating the overall concept of this processing architecture.

4.2 Processing Pipeline

As the processing pipeline consists of multiple processing steps that can be combined and exchanged, this section cover the iterative approach on providing item labels for later predictive analyses. Within each iteration the pipeline was modified based on the previous iteration's result.

4.2.1 Unify Names

The very first and most basic step is to unify the text format using simple rules. Using this approach, names with different capitalization (especially in product names), different punctuation (e.g., “o-Ring” and “O ring”), or redundant whitespace get unified names.

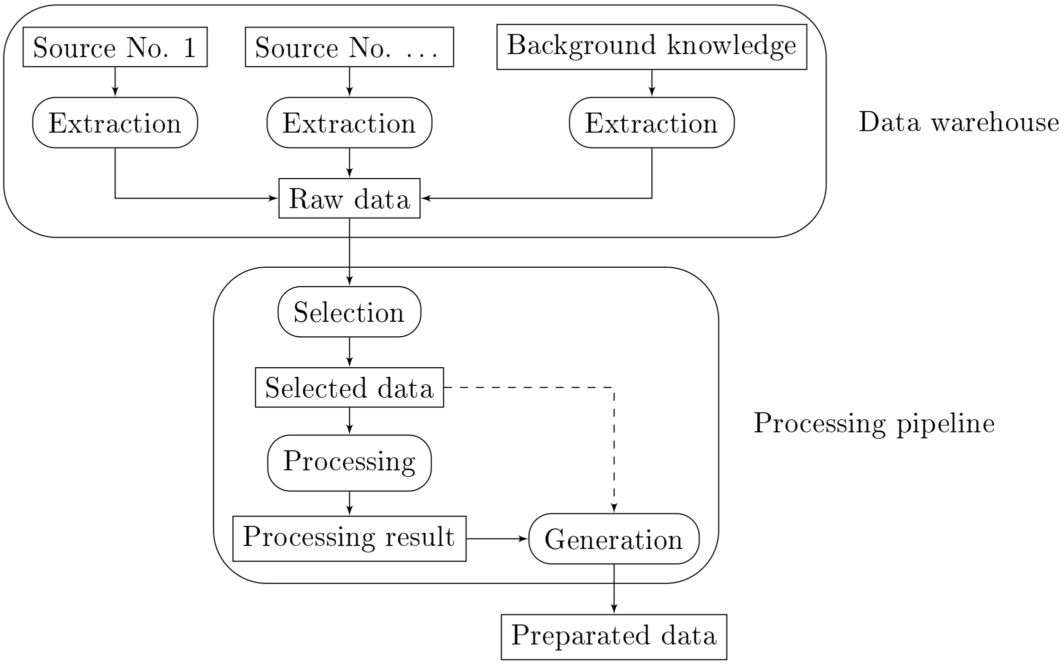

Figure 4.1: General architecture

This leads to a pipeline as shown in figure 4.2. Only the requisition item names are selected and used to generate the unified name.

For the unified name generation, these three rules provide a good start:

a) Reduce sequences of neither letters nor digits to a single space.

b) Trim leading and trailing whitespace.

c) Transform all capital letters to lower case.

Applying these rules, items that only vary in punctuation, capitalization, or whitespace get the same unified name. However, items with a relevant name variation, e.g., a different article number, remain distinct.

These unified names are no proper labels for model training but more sophisticated analyses become more easy.

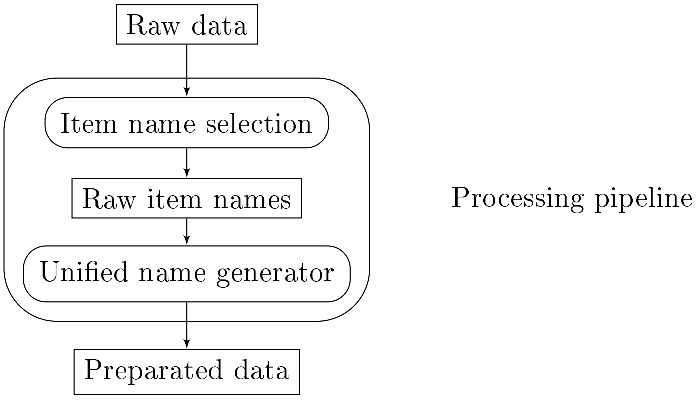

Figure 4.2: Processing pipeline: Item unification

4.2.2 Catalog Item Matching

Based on the unified names, the pipeline was extended to find the best matching catalog item for each requisition item, as shown in figure 4.3. In addition to requisition item names, catalog item names are unified using the same rules.

With both unified names available, various heuristics can be used to match requisition items to catalog items. The Hamming distance [Bookstein et al., 2002] was used due to its simplicity.

Analyzing the generated associations gave three interesting results:

a) A bunch of items could be associated with high precision. These requisition items all belong to the same catalog and seem to have been copied from a digital article list of that catalog.

b) Most of the other items don't really have an obviously matching catalog item.

c) Some special positions (e.g., "Freight Charge") are included in the item lists and are not flagged as such.

4.2.3 Special Positions

To get the found special position items out of the way, additional filters are needed. As the raw data set contains a list of special position names, these can be used as a blacklist. The updated pipeline is shown in figure 4.4.

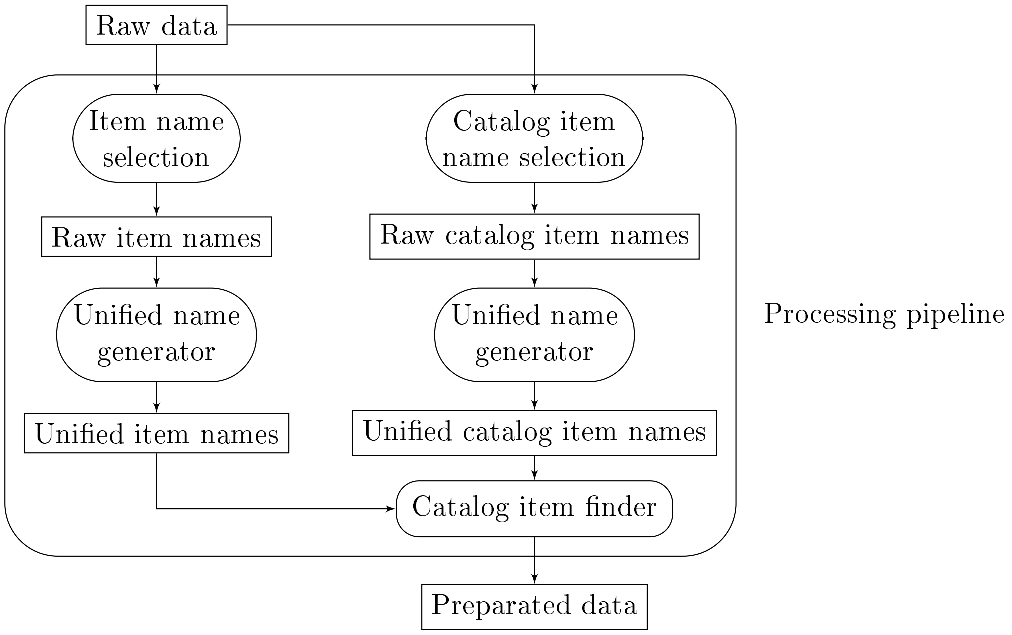

Figure 4.3: Processing pipeline: Catalog item matching

Filtering out special positions improve the overall result but the main problem still exists: Many requisition items don't have a directly matching catalog item.

4.2.4 Extracting Structure & Units

Matching requisition items based on simple text comparison only provided good result for a small subset of items. To get additional information from the item names, an analyzer for structure and units was added to the pipeline as shown in figure 4.5.

Simple structure and unit extraction based on a unit whitelist and text separators did not provide any useful information. The main problem with the item name's structure is that there is no obvious pattern in those names. Because of that no useful structure information is generated when simply using text separators.

Simplistic unit extraction is difficult too. On the one hand, units are case sensitive so the name unification has to be adjusted appropriately, on the other hand many items have units not clearly separated from their context (e.g., “120mmx60mm” or “w12cmh5cm”).

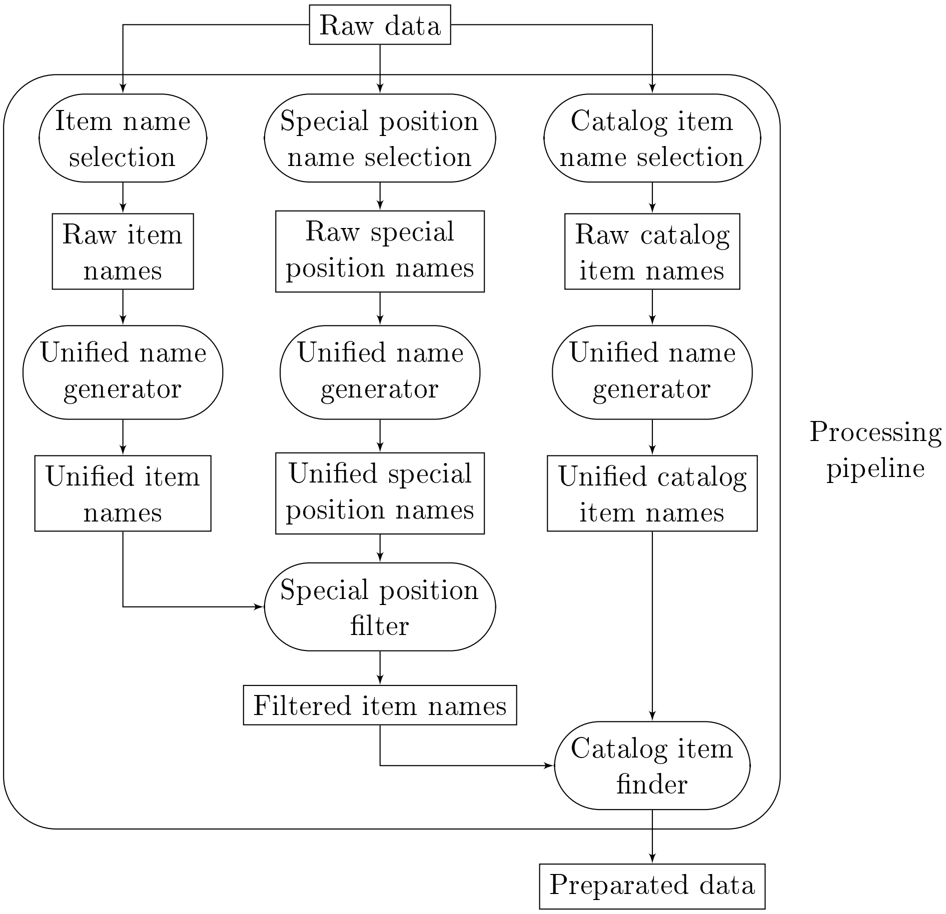

Figure 4.4: Processing pipeline: Special position handling

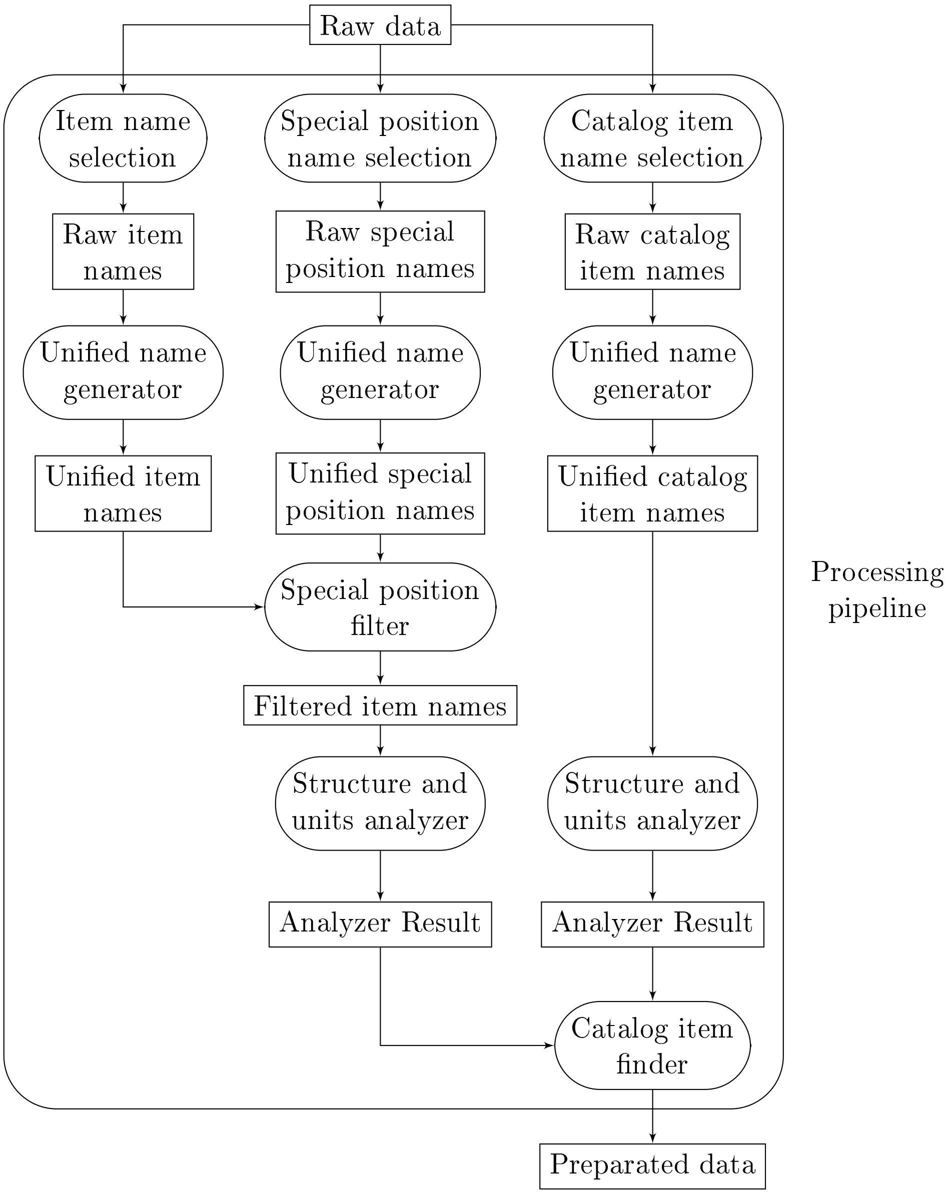

Figure 4.5: Processing pipeline: Structure analyses

Context sensitive structure and unit extraction using a LL(*) parser [Parr & Fisher, 2011] gave good results for a bunch of items. Due to a wide variety of structural differences, changing the grammar to handle all cases correctly is difficult and introduces high grammar complexity.

Unit handling, however, is not a problem with a proper grammar. There are only a bunch of useful unit contexts and simple grammar rules can be written for each case. But embedding these rules into the overall structure grammar is challenging as there are special cases that have to be handled correctly. Main issue here are article numbers within the item names that start with a number and end with a unit like suffix (e.g., “02341m”).

Matching catalog items with the created structure and unit information is also more complex than simply comparing strings. Searching for items with compatible units first helped to improve the overall result and performance, but comparing structure information is difficult. A different order of segments or subsegments is quite common and requires advanced structure tree comparison.

As this approach introduced quite a lot more complexity and each special case has to be implemented by hand, it's not practical.

4.2.5 Stopword Elimination & Part of Speech

Instead of matching requisition items to catalog items, another approach is to create an independent labeling and grouping. This can be achieved by generating a category tree based on the item names. Within this tree, each path to an item or subtree can be used as a label for everything within this path and each subtree provides an implicit item group. Having such a category tree implicitly allows primitive measurement of item similarity based on distance within the tree.

To generate such a category tree the processing pipeline has to be changed as shown in figure 4.6. The structure analyzer is changed to only handle simple sections based on punctuation. Within each section, remaining words are ordered based on part of speech information with more generic words (e.g., nouns) first. This gives an ordered list of words used to create the category tree.

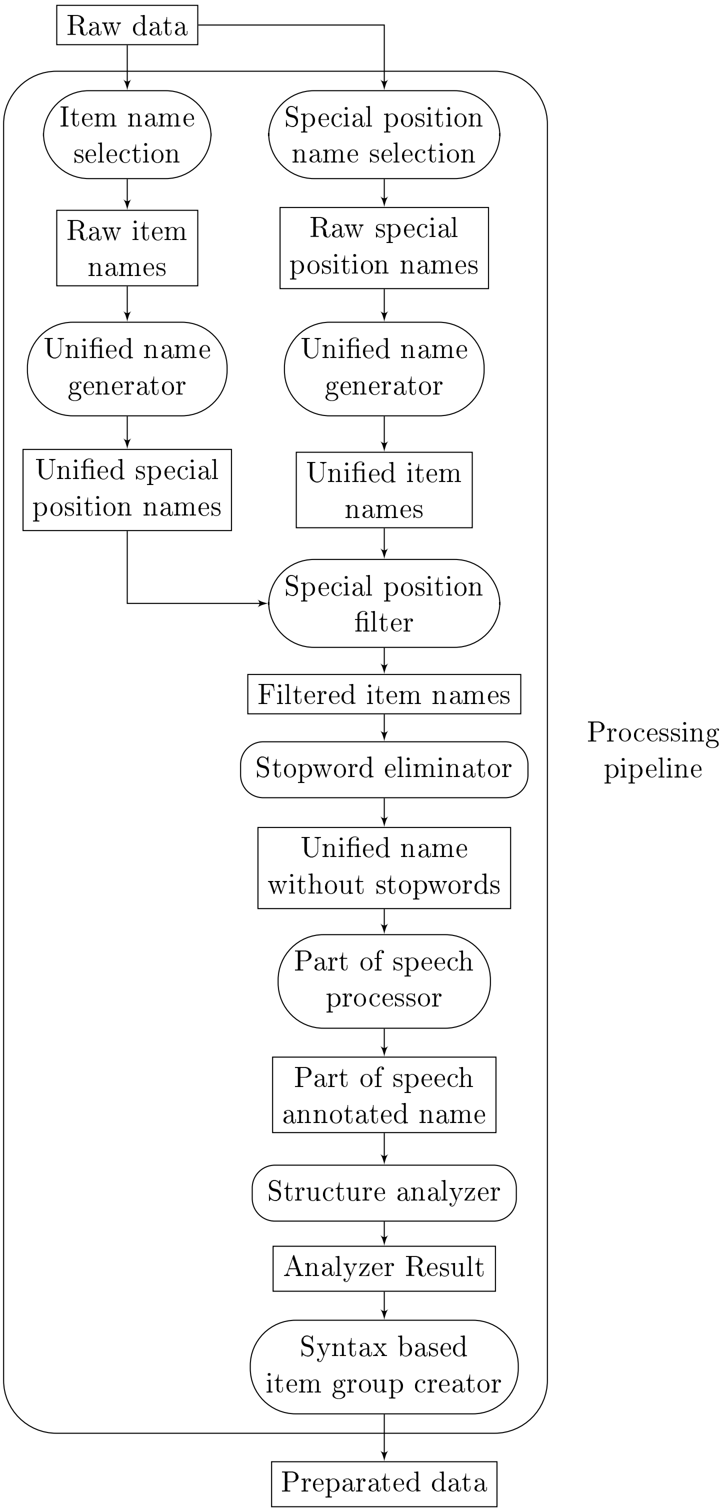

Figure 4.6: Processing pipeline: Part of speech processing

The generated category tree correctly groups similar items as expected. Different order of segments, however, is still a problem as these items are not grouped close together.

4.2.6 Removing Structure

To resolve the issues with different section ordering, the structure analyzer was removed as shown in figure 4.7.

With this change, only a few of the structure related grouping issues were resolved. There was no significant improvement of the generated category tree, many items are still not properly grouped together and many groups contain only single elements.

4.2.7 Bag of Words

Moving on to a bag of words [Harris, 1954] based approach, the processing pipeline is updated as shown in figure 4.8. The item name unification is also adjusted to be more strict. Unified names only consist of lowercase letters, digits and spaces.

The category tree is simply generated by splitting each unified name into *words* that only consist of lowercase letters and digits, separated by a number of spaces. From this sequence of words, all stopwords are removed. By creating a bag of words from all items, each sequence can be ordered by word occurrence in the selected data set. These ordered sequences now become the item's path within the category tree.

Using this approach gave quite good results. The top and intermediate level categories are useful and group all relevant items. However, a bunch of requisition items seem to be *completely* different from everything else. These items have unique, often single or two word, names and only occur once within the selected data set.

4.2.8 Different Bag of Words Sizes

To refine the previous result, changes to the bag of word configuration are made. Starting by adding a tiny bit of context to each entry by looking at word tuples of length one up to three, another adjustment option is to respect original word order within each tuple or simply transforming them into small word sets. Finally, word transformation can be

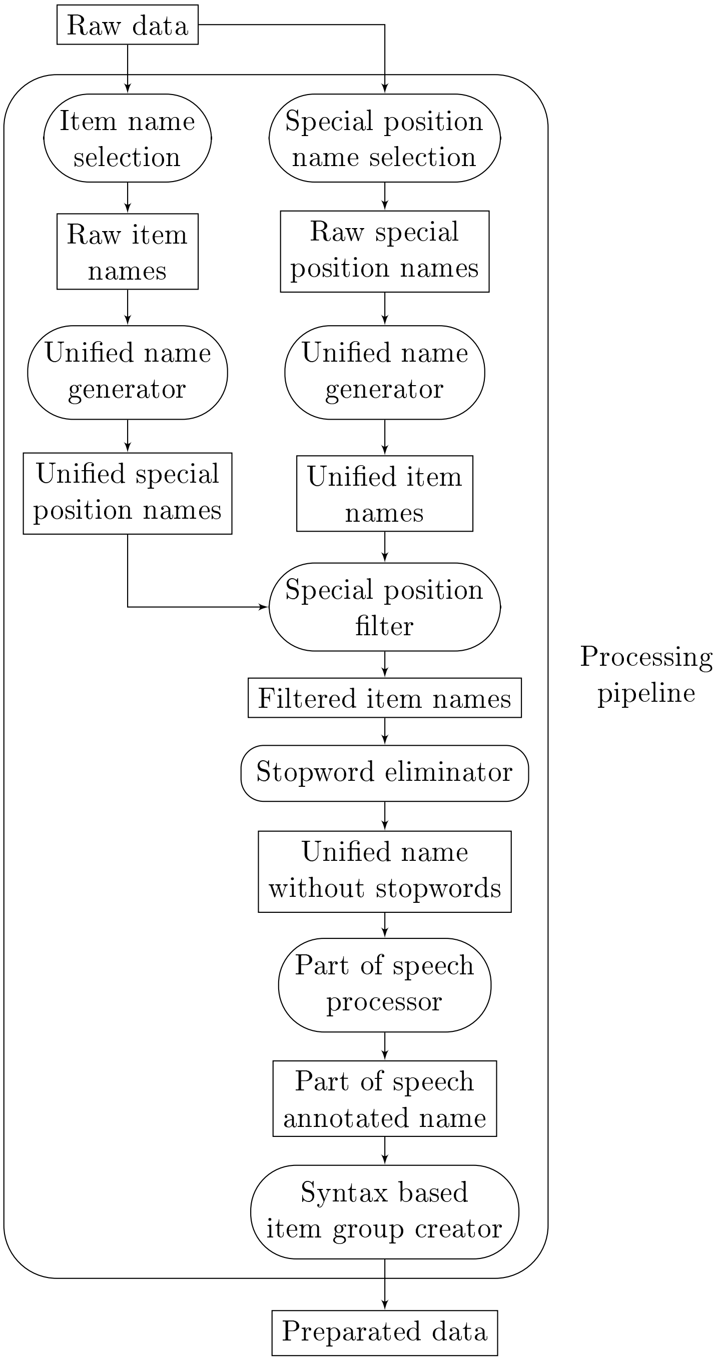

Figure 4.7: Processing pipeline: Ignoring structure information

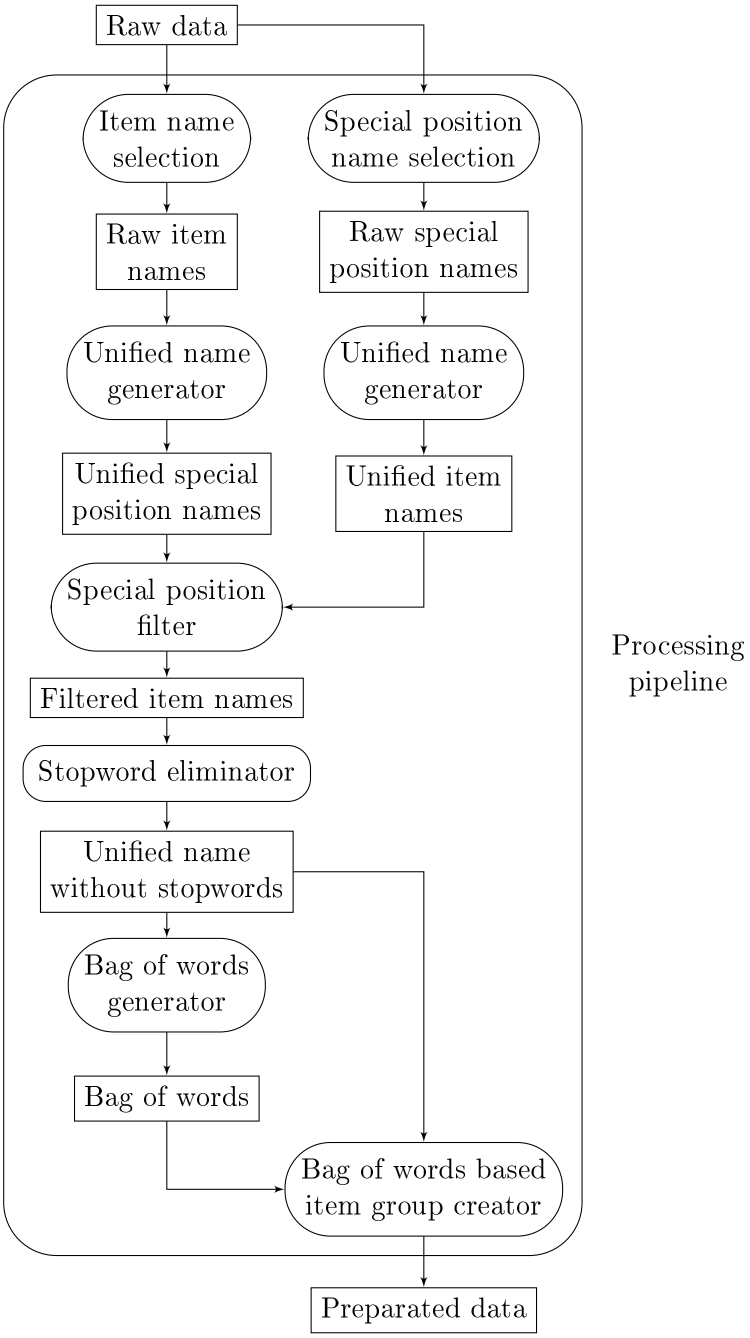

Figure 4.8: Processing pipeline: Bag of words

added to, e.g., reduce plural forms to singular ones or completely transforming each word to its word stem.

Adding context to the bag of words entries didn't have much impact of the generated category tree. Using a neighborhood of two for these entries gave the best results.

Ignoring the order of occurrence within each tuple had more impact. With this change, requisition items like "stainless steel" and "steel, stainless" get the exact same paths within the generated tree.

Reducing plural words to their singular form or using word stemming further improved the result as plural forms or other inconsistencies are sometimes used within item names.

4.2.9 Using Less Items

After fine tuning the category tree generation, the problem with unique items that are completely different to anything else, still remains. As these items are small in number and can be detected quite easily, they can be filtered out for later experiments. Those items don't matter for our overall goal to create a forecasting model.

For economical reasons, it is better to simply ignore them and to have the model forecast less items. The existing business process works without forecasting and even a model that only forecasts a few items improves that. Forecasting too many items, however, should be avoided as that results in additional costs and might hinder running business processes.

4.3 Summary

The used data set required specific preparation in order to perform predictive analysis. Item labels are missing in the data set and most items are simply identified by a free-text name. These free-text identifiers had inconsistent naming formats and their structure followed no obvious convention. In addition, different types of spelling errors can be found.

As proper item labels are required to perform predictive order analyses, they have to be generated based on the available information. The created item category tree provides such labels based on the paths within that tree. This also allows to extract grouping information in addition to the item labels.

Using an iterative approach gave quite good results in the end. The generated category tree provides useful top and intermediate level grouping that can be used as hierarchical labels for predictive analyses. Whether or not these labels allow a good order prediction has to be proven in future experiments. If such evaluation shows that there are still issues with labeling, further improvements are possible and can be implemented as needed.

All in all, the used iterative approach and the resulting architecture proved to be practical and is useful for similar problems. The evaluation driven iterations and the flexible architecture support a result oriented workflow.

4.3.1 Evaluation

Starting by simply trying to match requisition items to catalog items was useful as it helped to discover further data quality issues.

Getting too much into detail with sophisticated structure analysis, however, proved to be neither productive nor helped to discover relevant information.

The simplistic and seemingly stupid approach, neglecting all background knowledge about the item names, gave the best results.

In the end, the data quality was worse than expected and it required much work to get proper results.

4.3.2 Further Improvements

The result can further be improved by either changing the processing pipeline or using additional information from the data warehouse.

Despite creating a new processing pipeline, the segments itself can be improved. Exchanging the used algorithms or their configurations can easily be done without touching anything else. This also allows to directly compare their performance against each other.

Taking a look at the full order process up to the actual delivery of supplies to the vessel allows to perform ground truth analysis. Within this order process, the originally requested item can be replaced by an equivalent one. Using this knowledge allows us to verify the generated category tree in terms of item equivalence and provides more information for better item grouping.

In addition to that, quality of future data can be improved by helping users to use catalogs instead of free text items. This can be done by updating the item creation workflow. Our created category tree could be used to provide sophisticated suggestions for user input. Collecting data about the entered text and the accepted suggestion would also be valuable.

4.3.3 Generalization

For the item labeling problem, the business context is not relevant at all. It provides no useful information that can be used to improve label generation based on free text.

The structure analyses of item free text names can be used in any item classification problem that involves items with structured names. However, these structure analyses did not provide means to create proper item labels due to the structural complexity of the used item names. Too many variations and special cases were found. For data sets with more simple item names, these structural analyses might provide useful information.

The label generation based on a category tree can be used for grouping any short text snippets based on similarity. This text analysis ignores any sentence structure or grammatical information and only works based on word counting. Grouping longer text properly is difficult for this approach but any simple free text name classification works well with it.

4.3.4 Comparison with other Text Mining

Comparing the result to the Twitter and text analysis text mining approaches as described in chapter 2.4.4, similarities and differences can be found. While the texts used for mining are quite different (Tweets vs. item names), some heuristics and algorithms are used in both cases.

A review of different Twitter sentiment analysis techniques done by Bhuta et al. [2014] includes two relevant approaches.

The lexicon based approach is quite similar to our text grouping. The main difference is that we take the word's direct neighborhood into account but don't assign a positive or negative meaning to it. Topics and opinions are not relevant for us, as we are aiming to group similar items together.

In comparison to the Naïve Bayes approach, we neither use a probabilistic classifier nor a decision system. Probabilistic classifiers, however, could be a valid alternative to the bag of words approach.

Twitter text preprocessing done by Singh & Kumari [2016] is quite similar to our preprocessing. They are also using a preprocessing pipeline with similar steps. The two main differences are that we don't have a folksonomy that could be extracted and we don't need slang word replacement due to the nature of our item names.

4.3.5 Next Steps

Using the generated item category tree, the next steps towards an order prediction are to analyze the difference between requested and actually delivered supply items, and feature engineering to extract relevant information from the data categories from chapter 3.1.

To fully evaluate our result, it has to be tested within a prediction model, so we need to create such a model first.

5 Summary

Creating an order prediction requires data preprocessing and preparation first as there are usually issues that prevent working with the raw data. Some data might be missing and other data might be implausible within its context. Detecting these issues and finding a viable solution for them is crucial for a good prediction model. In addition, extracting further information using other statistic methods is usually needed for model training.

The preprocessing and preparation is usually done by following the general KDD workflow as described by Fayyad et al. [1996]. This approach separates the preprocessing and preparation into several incremental steps, starting with simple data selection.

For a proper data selection a good distribution of data aspects for entities to be predicted is required. This requires proper identification of entities, known as labels. These labels not only identify entities, they are also used within the prediction result.

If those entity labels are missing or otherwise not usable, a suitable replacement is needed. Such a replacement can either be found within additional data from other sources or can be extracted from other entity aspects using a mapping or approximation.

In the end, insufficient data quality is a huge issue for data analysis and data mining. If the amount of implausible or missing data is too large, it might be impossible to create a working prediction model and the data quality has to be improved by other means first.

5.1 Approach

After extraction of the data from multiple sources, the found entities have been categorized to get an overview. This grouping, as described in chapter 3.1, has been done based on business context aspects to identify primarily relevant parts.

The first relevant part of data was then selected and further analyzed in chapter 3.2. The aim of this analysis was to get an overview of existing data aspects and to find any missing or implausible data.

During this data analysis, missing labels have been found. The item identification provided by the data was inconsistent and not good enough to be used as labels within an order prediction. This is described in detail within chapter 3.4.

To create suitable labels, an approximation system based on text analysis has been developed in chapter 4. This system categorizes the found items into a tree structure, providing multi-level grouping and paths that can be used as item labels.

5.2 Data Labeling

Data labels provide identification for the labeled entities and can be described by an identification function. As this identification was not provided by the used data directly, it had to be calculated based on existing data.

Such a function could not directly be derived from other entity features, because of insufficient data quality of suitable entity fields.

However, approximating such an identification function is possible. Because the result is an approximation, it introduces an identification error. While the goal was to minimize this error, it still exists and can have a large impact, especially on later KDD steps.

5.3 Data Quality

Within data mining insufficient data quality can cause big issues.

Missing data, on the one hand, can occur often enough so it is not possible to simply drop out such entities, and reconstructing missing values can be hard. Additionally, reconstructing data will introduce some error.

Implausible data, on the other hand, can be hard to detect. Reference data from other sources or expert knowledge might be needed to identify such cases. Dealing with implausible data is done in the same way as with missing data and therefore causes the same problems.

If the overall data quality is too bad, data mining might not be possible. The amount of remaining complete and correct data might be to small for proper model training or the error introduced by synthetically generating replacement values might be too large.

In such a case, working on the data quality itself is required. For this, the system generating the data has to be improved. This can be done by either changing technical or implementation details, or by improving the user interface of the system. After that, the new data can be used to start the data analyses and data mining process anew.

Bibliography

[Anders 2018] ANDERS, Lucas: *Deep Learning zur Vorhersage des Energiebedarfs der antarktischen Forschungsstation Neumayer III*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelor thesis, October 2018. `https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/anders.pdf` 3, 5

[Bhuta et al. 2014] BHUTA, S. ; DOSHI, A. ; DOSHI, U. ; NARVEKAR, M.: A review of techniques for sentiment analysis Of Twitter data. In: *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2014, pp. 583–591 6, 28

[Bogina et al. 2016] BOGINA, Veronika ; KUFLIK, Tsvi ; MOKRYN, Osnat: Learning Item Temporal Dynamics for Predicting Buying Sessions. In: *Proceedings of the 21st International Conference on Intelligent User Interfaces*. New York, NY, USA : ACM, 2016 (IUI '16). – ISBN 978–1–4503–4137–0, 251–255 5

[Bookstein et al. 2002] BOOKSTEIN, Abraham ; KULYUKIN, Vladimir A. ; RAITA, Timo: Generalized hamming distance. In: *Information Retrieval* 5 (2002), No. 4, pp. 353–375 16

[Council of the European Union 2016] COUNCIL OF THE EUROPEAN UNION: Commission Implementing Regulation (EU) 2016/1927 of 4 November 2016 on templates for monitoring plans, emissions reports and documents of compliance pursuant to Regulation (EU) 2015/757 of the European Parliament and of the Council on monitoring, reporting and verification of carbon dioxide emissions from maritime transport (Text with EEA relevance). In: *Official Journal of the European Union* 59 (2016), November, 1-21. `https://eur-lex.europa.eu/legal-content/GA/TXT/?uri=CELEX:32016R1927`. – ISSN 1977–0677 4

[Fayyad et al. 1996] FAYYAD, Usama ; PIATETSKY-SHAPIRO, Gregory ; SMYTH, Padhraic: From data mining to knowledge discovery in databases. In: *AI magazine* 17 (1996), No. 3, 37. `https://www.aaai.org/ojs/index.php/aimagazine/article/download/1230/1131/0` vi, 4, 29

[Harris 1954] HARRIS, Zellig S.: Distributional structure. In: *Word* 10 (1954), No. 2-3, pp. 146–162 22

[Inmon 1996] Inmon, William H.: The data warehouse and data mining. In: *Communications of the ACM* 39 (1996), No. 11, pp. 49–51 14

[Lind et al. 2015] Lind, Mikael ; Brodje, Anders ; Haraldson, Sandra ; Hagg, Mikael ; Watson, Richard: Digitalisation for sustainable sea transports. Version: 2015. `http://dx.doi.org/10.1049/PBTR001E_ch9`. In: *Clean Mobility and Intelligent Transport Systems*. Institution of Engineering and Technology, 2015 (Transport). – DOI 10.1049/PBTR001E_ch9, Chapter 9, 187-218 2

[Loper & Bird 2002] Loper, Edward ; Bird, Steven: NLTK: the natural language toolkit. In: *arXiv preprint cs/0205028* (2002) 12

[Parr & Fisher 2011] Parr, Terence ; Fisher, Kathleen: LL(*): The Foundation of the ANTLR Parser Generator. In: *SIGPLAN Not.* 46 (2011), June, No. 6, 425–436. `http://dx.doi.org/10.1145/1993316.1993548`. – DOI 10.1145/1993316.1993548. – ISSN 0362–1340 20

[Singh & Kumari 2016] Singh, Tajinder ; Kumari, Madhu: Role of Text Pre-processing in Twitter Sentiment Analysis. In: *Procedia Computer Science* 89 (2016), 549 - 554. `http://dx.doi.org/https://doi.org/10.1016/j.procs.2016.06.095`. – DOI https://doi.org/10.1016/j.procs.2016.06.095. – ISSN 1877–0509. – Twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India 7, 28

Glossary

atomizer Device that breaks bulk liquids into small droplets, creating aerosols.

folksonomy The spontaneous cooperation of a group of people to organize information into categories; the practice and method of collaboratively creating and managing tags to annotate and categorize content; a user-generated taxonomy.

o-ring A mechanical seal in shape of a torus. Also known as packing or toric joint.

requisition Request for supplies issued by the ship. The office then reviews the request, orders missing parts and organizes the delivery to the vessel.

vessel Nautical term for ship.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: ________________________________

Vorname: ________________________________

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Data analyses and preparation for machine learning based order prediction

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

____________ ____________ ____________________

Ort Datum Unterschrift im Original