Baltic J. Modern Computing, Vol. 11 (2023), No. 3, 475-499
https://doi.org/10.22364/bjmc.2023.11.3.08

Data Warehouse Data Model Improvements from Customer Feedback

Jānis ZEMNICKIS

University of Latvia, Faculty of Computing, Raiņa bulvāris 19, Riga, LV-1586, Latvia

janiszemnickis@gmail.com

ORCID 0009-0002-4388-8428

Abstract. The amount of unstructured data in organisations is growing each year. Unstructured data could hide huge amounts of information, which could improve an organisation's performance and daily processes. In comparison to structured data, unstructured data is more complex to analyse and get useful information from due to a lack of competence and experience in organisations. Big data technologies open new opportunities for unstructured data processing and for later analysis. Organisations build data warehouses for the purpose of data analysis. A data lake, which is implemented using big data technologies, is used as one of the data sources for data warehouses. Meanwhile the data warehouse and its model creation could be a challenging task in order to meet all stakeholders' expectations and support all business use cases. There are known approaches on how to develop data warehouse data models by relating the organisation's key performance indicators (KPI) to information requirements, but in this article a new method is introduced which allows one to extend the data warehouse data model using unstructured data in a semi-automated way. This method is designed for the purpose of processing client feedback in the field in which the organisation operates in order to acquire relevant aspects about the organisation's operations, in other words quantitative KPIs. From the acquired KPI information, the requirements of the data warehouse are obtained, which are transformed into the attributes of the data model of the data warehouse. Newly acquired attributes are integrated into the existing data model of the organisation's data warehouse. The method can be divided into four phases: 1) data gathering – streaming and storing data on server 2) calculations – natural language processing (NLP) and special calculations to prepare data for KPI generation 3) KPI generation – phase where new KPIs and KPI values are defined using special algorithms 4) data model improvements – phase where new data model elements are discovered and integrated into the existing warehouse data model.

Keywords: Data warehouse, data model, big data, natural language processing, key performance indicators.

1. Introduction

The purpose of a data warehouse is to provide significant information to decision makers and ensure data analysis opportunities. A data warehouse can contain different information from various data sources (Bimonte et al., 2020). Bill Inmon defined (Inmon, 2005) a data warehouse as "a subject-orientated, integrated, time-variant and non-volatile collection of data to support the decision-making process". During recent years, data warehouses have been considered as the most effective tool for supporting decision-making (Benkhaled and Berrabah, 2019). In a data warehouse, data is usually stored in a special denormalized multidimensional data model, which consists of cubes and dimensions. Usually, facts and measures are stored in cubes, but in dimensions, attributes related to these facts. Attributes can form a hierarchy. Nowadays, data volumes continue to rise due to new technologies, social media and opensource data (Bimonte et al., 2020). There is an opinion that traditional data warehouses that use structured data as a data source do not meet the growing needs of modern organisations, such as processing different types of data from social networks, sensor data (Bouaziz et al., 2019). Additional data and data types open the possibility for extensive analysis possibilities to reveal information important for the development of the organisation. There is a significant difference between traditional requirement engineering for operational information systems and data warehouses (Prakash and Prakash, 2019). Developing a data warehouse can be a challenging task, especially for organisations with no prior experience in data warehouse development. Special attention is paid to data warehouse development requirements (Prakash and Prakash, 2019). Despite the fact that data warehouses generally offer a range of benefits to organisations, data warehouse development can fail if it does not provide stakeholders with the information they need (Benkhaled and Berrabah, 2019). The cost of developing a data warehouse is usually high, due to the need for a highly skilled workforce. One of the reasons that can be attributed to data warehouse failures is a data warehouse data model that does not meet the business interests and needs of the organisation. The authors (Bimonte et al., 2020) propose to divide the data warehouse data model development methodologies into three groups:

- Data-driven approach, based on data source database schema analysis
- Requirements-driven, methods in which data model requirements are obtained from users
- Mixed, combines a requirements-driven approach with a data-driven approach

The purpose of this article is to introduce a new method on how to improve data warehouse data model by analysing unstructured data. This method belongs to the mixed group, because the elements of the data model are generated using data from the source systems, but it is expected that the data model requirements of the obtained data warehouse need to be validated with business representatives. The method involves analysing the feedback of the organisation's customers with big data and NLP technologies in order to obtain customer ratings for a specific field and service. The obtained information is used in special calculations, which result in:

- KPIs relevant to the operation of the organisation
- Optimal values of these KPIs
- New data warehouse data model attributes

The rest of the article is organised as follows: In section 2, a review of technologies and KPIs. In the 3rd section there is a review of related work. In section 4, a new method to improve the data warehouse data model from user feedback is introduced. In section 5 the method's use case and results are presented. Section 6 discusses a review of possible future work, conclusion, evaluation of method and method limitations.

2. Background

In this section, technologies which are used by the method, key performance indicators and data warehouse requirements are described.

2.1. Information requirements for data warehouses

Data warehouse data model should be designed according to data warehouse information requirements to meet stakeholders' expectations (Winter and Strauch, 2003). Information requirement gathering for data warehouses could be significantly different from standard operational system requirement gathering (Prakash and Prakash, 2019). Information requirements for data warehouses could be extracted by using several methodologies: a) data-driven, b) goal-driven, c) user-driven (List et al., 2002). The organisation's KPI analysis is also one of the methodologies which can be used to gather information requirements for data warehouses (Kozmina, 2017) and (Niedritis, 2011).

2.2. KPI and KPI monitoring

Organisations that perform KPI determination and evaluation often use a data warehouse as a process measurement and monitoring system (PMMS). KPIs are widely used in organisations to measure their performance. KPI characteristics and features are related to terms such as success factors and performance measures. The authors (Parmenter, 2015) describe success factors as industry-defined aspects that determine an organisation's performance, while critical success factors are those aspects or problems that characterise the basic state of the organisation, i.e., the most important factors of the organisation's operation. Performance measures mean measurements that characterise performance in the organisation. In cases where data warehouses are used in organisations to monitor KPI values, the data model of the organisation's data warehouse should be created according to the KPI calculation needs. KPIs are indicators that are most important, or those characterised by critical success factors (CSFs). The author (Parmenter, 2015) claims that "KPIs tell you what to do to increase performance dramatically".

One of the properties of a KPI is its hardness. KPIs can be "soft" or "hard" (Domínguez, 2019). If a KPI is "soft" then it is not directly measurable, it is qualitative, for example an employee's or companies' reputation. If a KPI is "hard" then it can be directly measurable; this KPI type is quantitative, for example the number of sales.

2.3. Big data

With the evolution of technologies such as cloud computing technologies, 5G, Internet of Things, the amount of data in organisations is growing rapidly. Big data analysis technologies are a key aspect in the intelligent processing of heterogeneous data (Wang et al., 2021). Enterprises are using big data technologies which are getting more popular. In the period from 2011 to 2022, the most popular industries in which big data analytics are applied are engineering, computer science, automation control systems, business economics, telecommunications (Wang et al., 2021). Big data analytics is often implemented by building data warehouses on top of big data. In an enterprise, a data lake is used as a data warehouse source system, which is often implemented using big data technologies. Google[1] enterprise has defined a data lake as "a centralised repository designed to store, process, and secure large amounts of structured, semi-structured, and unstructured data. It can store data in its native format and process any variety of it, ignoring size limits."

Big data can be described with these three 3 'V' characteristics: Volume, Variety, and Velocity. These Vs characterise the large amount and complexity of data as well as the data generation speed that causes the necessity to deal with the streaming data. Other Vs were added to the list later, for example, veracity and value, which describe the uncertainty and business value of big data. The arrival of big data in data warehouse projects places new requirements on the traditional DW systems (Benkhaled and Berrabah, 2019). Big data are often unstructured or semi-structured. Research and practice indicate that such information can be interesting for the decision-making process (Pejić Bach et al., 2019). According to the authors (Danaher et al., 2015) organisations only analyse 12% of their available unstructured data. 95% of organisations struggle with effective unstructured data analysis (Baviskar et al., 2021). Almost 80% of organisations have no or some understanding of what's happening with their unstructured data (Rizkallah, 2017). 87% of businesses recognise the disadvantage of not completely utilising the full scope of the available data due to the inability to derive value from unstructured data (Howatson, 2016). The problems faced in extracting knowledge from unstructured data means that organisations that regularly consume large amounts of resources encounter data analysis problems (Briggs and Hodgetts, 2017). A study shows that 80% of data in organisations is unstructured (Rizkallah, 2017). Analysis of unstructured data is more complicated compared to the analysis of structured data (Howatson, 2016). There are still many undiscovered opportunities in the analysis of unstructured data (Pejić Bach et al., 2019). It follows that organisations have far more unstructured data than structured data. However, there is a lack of competences and experience to process unstructured data, which leads to project failure without reaching business value and overbudgeting the project; meanwhile, business representatives see that there is huge potential in analysing unstructured data to discover hidden value to improve an organisation's performance.

2.4. Natural language processing

Natural human language is part of unstructured data. This unstructured data can be processed using natural language processing (NLP) technologies. Authors (Chowdhary and Chowdhary, 2020) define NLP as follows: "Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things". One of the features of artificial intelligence is NLP, which has potential in processing large amounts of natural language in order to acquire data in a computer understandable format (Garg et al., 2021). NLP reduces the communication barrier between the human and computer because NLP offers tools and functionality for the computer to understand human language. A study (Zhao et al., 2021) shows that the most frequently used NLP techniques are POS tagging, Tokenization, Parsing, Stop-Word Removal. These techniques could be implemented by using tools or libraries such as Standford CoreNLP, NLKT, Apache OpenNLP, SparkNLP. NPL is used in various industries, such as

[1] https://cloud.google.com/learn/what-is-a-data-lake

information technology, robotics, artificial intelligence, electronic engineering, etc. By applying NLP to unstructured data, which is obtained from a social network such as Twitter, Facebook or Instagram, valuable information can be extracted, for example, by performing sentiment analysis (Garg et al., 2021). By implementing NLP tools in the organisation's IT systems and extracting useful information it is possible to improve customer satisfaction with the organisation's service or its operation as a whole.

3. Related work

There are known data warehouse data model development methods (Kaldeich and Sá, 2004), (Winter and Strauch, 2003), (Wang, 2008). The popularity and use cases of NLP analysis are growing each year (Zhao et al., 2021). A review of related studies shows that there is no existing method for how to extract data warehouse attributes from user feedback.

Similar to the method proposed by the author in the article (Pernici et al., 2018), KPI values are determined using big data technologies, but the determination of new KPIs and the inclusion of the attributes necessary for KPI calculations in the data model of the data warehouse of the data organisation are not ensured.

In the article (Navinchandran et al., 2021), user Work Orders have been analysed in order to determine existing KPI values. The Work Order contains free text which is submitted by IT system users. In order to set KPI target values, NLP analysis has been applied. This method does not provide a solution to set new KPIs.

Another method (Doshi et al., 2016) does not provide further processing of the obtained metadata, such as integration into a data model or compilation of KPIs.

The method (Bouaziz et al., 2019) describes a technique for developing a data warehouse schema using schema free databases such as NoSQL databases. The first step of the method is to obtain the structure of each database record; it is important to do so for each record, as it can vary. The next step of the method is to identify the structure of the graph. The graph is created because it helps the designer to identify the data warehouse schema's multidimensional concepts. The next steps are to identify multidimensional concepts and design the data warehouse schema. Unlike the method described by the author of this article, it is intended to use schema free databases and not to process user reviews – natural language.

4. Method to improve the data warehouse model

This chapter describes the operation of the method and the corresponding use case examples. In this chapter, each subsection includes use case steps from the use case, which is described in section 5; such organisation of the paper is used to better describe the method. Fig. 1 shows a simplified diagram of the method. The method can be divided into eight steps. The first step of the method is "Data gathering" – obtaining and saving data on the server so that they can be used in the next steps of the method. The next step is "NLP processing"; the data read in this step is processed with an NLP tool to obtain word interdependencies in the text and sentiment classification; the results are saved on the server. The "Calculations" step calculates the most frequently used words and their dependencies with some measurements from text. In the step "KPI themes", the KPI themes are clarified, which will be used in the later stages of the method in the compilation of the KPI. The "Quantitative KPI values" are calculated in the step "Quantitative KPI values". In the "KPI conditions" step, the conditions for each KPI theme are manually defined. In the "KPI generation" step, KPIs are compiled from the data obtained in the previous steps. The final step of the method is "DW model improvements", which involves integrating the acquired attributes into the DW data model.

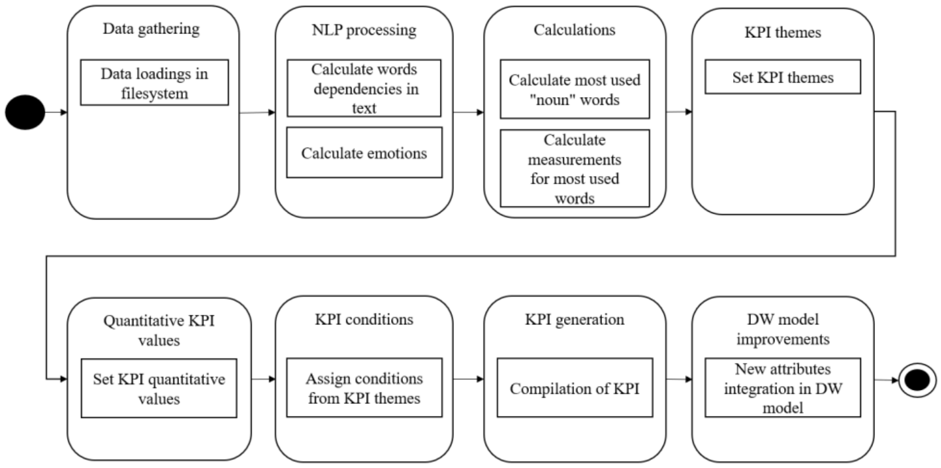

Data gathering
Data loadings in filesystem
NLP processing
Calculate words dependencies in text
Calculate emotions
Calculations
Calculate most used "noun" words
Calculate measurements for most used words
KPI themes
Set KPI themes
Quantitative KPI values
Set KPI quantitative values
KPI conditions
Assign conditions from KPI themes
KPI generation
Compilation of KPI
DW model improvements
New attributes integration in DW model

Fig. 1. Simplified method diagram

4.1. Data gathering

This method uses user feedback and comments as a data source. This feedback and comments are usually unstructured data. By analysing customer feedback, it is possible to acquire important information for the development of the organisation.

There are several possibilities that can be used in the organisation as a source of data to obtain feedback and comments: a) customer surveys, b) listening to complaints/suggestions, c) feedback about the organisation or industry outside the specific organisation, d) social media, for example Twitter data. It is possible to combine these data sources to obtain results based on a larger amount of data, which could provide better result data. Unstructured data in the data source can be related to any topic, and they can be unrelated to the specific industry that is being analysed; the method determines that analysis of data not related to the industry may have a negative impact on the results, therefore it is necessary to filter unstructured data and only use those related to the specific industry. In order to filter out the data corresponding to the specific industry, it is necessary to apply filters.

Use case example of data gathering phase

The use case example uses publicly available Twitter data as a data source for customer feedback. The method states that the data is collected for a specific field, in the specific example about catering, that is, the tweet should be related to the words “food”, “restaurant’ or “sushi”. The specific field and keywords have been chosen because the company in the use case example operates in the field of catering, providing catering services on site – in a restaurant and managing orders with delivery outside the restaurant. The main focus of the restaurant is Japanese cuisine – sushi. These words can differ in each organisation and for the corresponding field; the most suited keywords should be chosen according to the specifics of the field. Twitter tweets are interpreted as feedback or comments about a field, but other data sources can also be used in the method, such as YouTube, Facebook comments or customer surveys of the organisation.

4.2. NLP processing

The method involves the use of one of the natural language processing libraries (StandfordCoreNLP, NLTK, SparkNLP, etc.), which allows one to determine the coefficients of emotions for the text and the dependencies of the words in the sentence. Fig. 2 shows a general representation of the step of the method “NLP processing”. It is designed to process unstructured data that has been read into “Data storage” in the previous step of the method. Various technologies can be used as data storage (HDFS, Google Cloud BigQuery, Databricks Lakehouse Platform). Fig. 2 shows the elements “Dependencies calculation” and “Word dependency data”, which represent the calculation of the dependencies of words in a sentence. The elements “Emotion$_1$ calculation” and “Emotion$_n$ calculation” represent the calculation of various emotion coefficients. This information is stored in the data storage so that it can be used in the further stages of the method, see the elements “Emotions$_1$ data” and “Emotions$_n$ data”. In the author’s proposed method the use case of utilising the text is analysed according to three types of emotions – “sentiment”, “positive emotion” and “sarcasm”, but if the selected natural language library allows, then additional types of emotions by which to analyse the text can be added. In the later stages of the method, each additional type of emotion can provide additional information.

After this step of the method, there is access to information about:

- From “Emotion data” – positive or negative shade of text in one or more types of emotion
- From the “Word dependency data” calculation – a word and its word class in a sentence (for example, the word “potatoes” and the type “NNS – Noun, plural”)
- From the “Word dependency data” calculation – mutual relations of words in a sentence (for example, a very simple sentence “Warm potatoes”, the words “warm” and “potatoes” are connected by the link “amod – adjective determiner”)

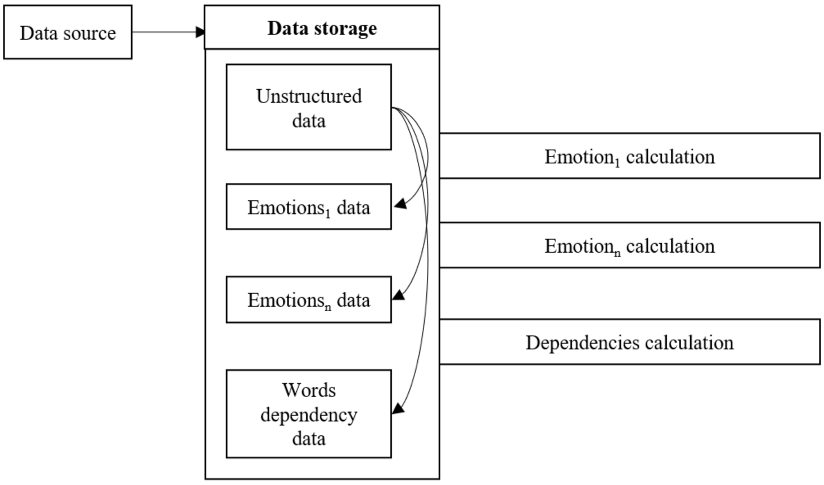


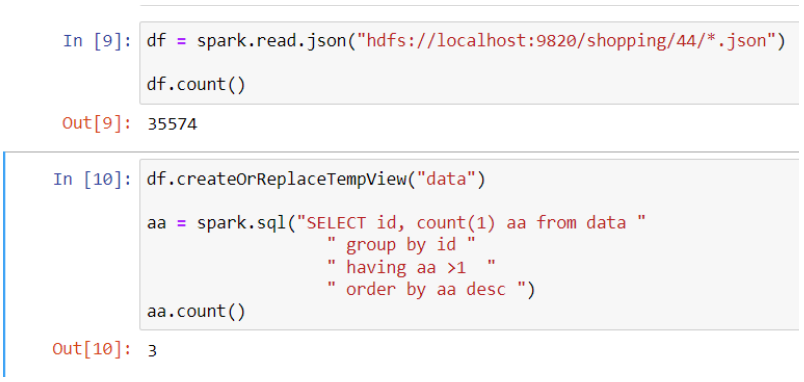
Fig. 2. NLP processing step of the method

Use case example of NLP processing phase and technical implementation

The use case example of the method is implemented using a set of multiple technologies. The Python library Tweepy is used to stream data from Twitter. Raw Tweets are stored on the server in JSON file format. The next step is to read these unprocessed Tweets into the Hadoop Distributed File System (HDFS), because the method requires process data using big data technologies. Apache Spark Engine is used to retrieve and process data quickly and efficiently from HDFS. The author uses the Jupyter project to be able to use the interactive, effective computing product Jupyter Notebook, which allows one to use a convenient graphical interface when developing Spark jobs. The Apache Spark module pyspark.sql is used in order to be able to develop an Apache Spark job in SQL syntax. Fig. 3 shows an example of how Apache Spark jobs are created from Jupyter Notebook. The first Apache Spark job with number 9 shows that all files with the data type JSON are read from HDFS and read into the variable “df”, which has a special Apache Spark data type DataFrame. The command df.count() counted and printed the number of lines read. The second Apache Spark job with the number 10 creates an Apache Spark temporary view from the resulting “df” named “data”. Later, this view can be used when writing an SQL query. A trivial SQL command is used to find duplicate IDs.

SparkNLP open source natural language processing library is used to process Tweet text. SparkNLP is used because it has a wide availability of open source models, is easy to install, and is widely used in organisations. SparkNLP is built on Spark ML as a foundation. SparkNLP annotator uses rule-based algorithms, Tensorflow, machine learning to implement deep learning. The Apache Spark ML solution allows for easy solution scale without code changes. Tensorflow allows one to use deep learning algorithms by using GPU (nVidia’s DGX-1 and Intel’s Cascade Lake processors). Spark NLP provides the most familiar NLP tasks – tokenization, lemmatisation, stemming, part of speech tagging, sentiment analysis, spell checking, named entity recognition, etc.

Spark NLP provides an opportunity to use publicly available open-source models that can be trained on your own data, as well as pre-trained pipelines and models. The author uses pre-trained models in the use case examples, which are used to find out word dependencies in sentences and text emotions, see the previous Fig. 2. Pre-trained models allow one to download the model at the time of its use, specifying the name and configuration. The pre-trained models of this SparkNLP process the free text of a tweet, for example the tweet “Had lunch over zoom some days ago. I guess I may open a restaurant at long last.”



```
In [9]:  df = spark.read.json("hdfs://localhost:9820/shopping/44/*.json")

         df.count()

Out[9]:  35574

In [10]: df.createOrReplaceTempView("data")

         aa = spark.sql("SELECT id, count(1) aa from data "
                        " group by id "
                        " having aa >1  "
                        " order by aa desc ")
         aa.count()

Out[10]: 3
```

Fig. 3. Example of Apache Spark job in SQL syntax

The use of SparkNLP pre-trained model classifierdl_use_emotion[2], which provides the functionality to determine the emotion of a sentence – surprise, sadness, excitement or fear. This SparkNLP model is trained using several datasets such as YouTube comments, Twitter and ISEAR (International Survey on Emotion Antecedents and Reactions) dataset. The resulting emotion coefficients are useful for analysing user feedback. A coefficient is available for each emotion, see

Table 1, where the tweet is processed with the classifierdl_use_emotion model. We can see that the SparkNLP classifierdl_use_emotion model has determined an example tweet “Had lunch over zoom some days ago. I guess I may open a restaurant at long last.” most closely matches the emotion of excitement with a coefficient of 0.9977035.

Table 1. Results of example tweet after applying the pre-trained model “classifierdl_use_emotion”

Emotion	**Coefficient**
Sadness	0.0006125135
Surprise	0.0016160638
Fear	0.0000679279
Excitement	0.9977035

[2] https://demo.johnsnowlabs.com/public/SENTIMENT_EN_EMOTION/

The use of SparkNLP pre-trained model SENTIMENTDL_USE_TWITTER[3], which uses the “Universal Sentence Encoder” (Cer et al., 2018) model as the base. This model is trained on a Twitter dataset consisting of 1.6 million tweets. Processing text with this model yields the odds of whether this tweet is positive or negative. See Table 2 for the example tweet. The example shows that the tweet is almost 100% positive.

Table 2. Results of example tweet after applying pre-trained model “SENTIMENTDL_USE_TWITTER”

Emotion	**Coefficient**
Negative	0.0003254917
Positive	0.99967456

The use of the SparkNLP pre-trained model “classifierdl_use_sarcasm”[4]. This model is trained using the Sarcasm-Detection dataset. This model determines whether the entered text is used as sarcasm. In the example shown in Table 3, it can be observed that the example tweet is not in a sarcastic sense, as it is almost 100% normal.

Table 3. Results of example tweet after applying pre-trained model “CLASSIFIERDL_USE_SARCASM”

Emotion	**Coefficient**
Negative	0.0003254917
Positive	0.99967456

The use of “dependency_typed_conllu” model[5]. This model splits the text into words and returns the part of speech (POS) [6] type of each word, the related word, and the type of that relationship. This model is trained using the “CONLL” dataset. When processing the Tweet used as an example “Had lunch over zoom some days ago. I guess I may open a restaurant at long last.” a result is obtained, which can be seen in Fig. 4, where information about each word in the sentence can be seen. The words from the example sentence are shown in the “chunk” column; the “begin” and “end” columns show the beginning and end positions of these words in the sentence. The POS column shows the

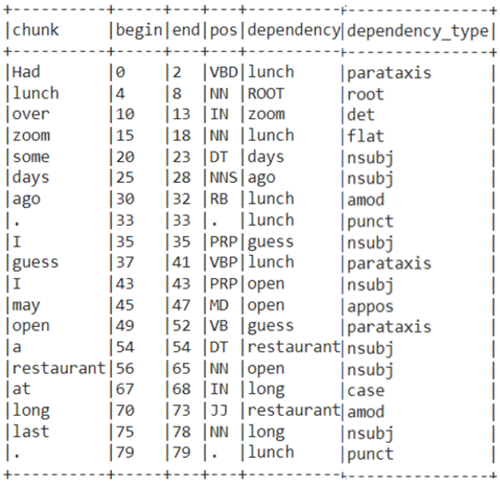
[3] https://nlp.johnsnowlabs.com/2021/01/18/sentimentdl_use_twitter_en.html

[4] https://nlp.johnsnowlabs.com/2020/07/03/classifierdl_use_sarcasm_en.html

[5] https://nlp.johnsnowlabs.com/2021/03/27/Typed_Dependency_Parsing_en.html

[6] https://towardsdatascience.com/part-of-speech-tagging-for-beginners-3a0754b2ebba

types of words in the sentence. The “dependency” column shows the dependent words, and the “dependency_type” column shows the dependency type of those words. For example, the word “restaurant” is in the sentence starting from position 56 to 65, its part-of-speech type “NN – noun”. This word is related to the word “open” with the relation type “nsubj – nominal subject”.



```
+----------+-----+---+---+----------+---------------+
|chunk     |begin|end|pos|dependency|dependency_type|
+----------+-----+---+---+----------+---------------+
|Had       |0    |2  |VBD|lunch     |parataxis      |
|lunch     |4    |8  |NN |ROOT      |root           |
|over      |10   |13 |IN |zoom      |det            |
|zoom      |15   |18 |NN |lunch     |flat           |
|some      |20   |23 |DT |days      |nsubj          |
|days      |25   |28 |NNS|ago       |nsubj          |
|ago       |30   |32 |RB |lunch     |amod           |
|.         |33   |33 |.  |lunch     |punct          |
|I         |35   |35 |PRP|guess     |nsubj          |
|guess     |37   |41 |VBP|lunch     |parataxis      |
|I         |43   |43 |PRP|open      |nsubj          |
|may       |45   |47 |MD |open      |appos          |
|open      |49   |52 |VB |guess     |parataxis      |
|a         |54   |54 |DT |restaurant|nsubj          |
|restaurant|56   |65 |NN |open      |nsubj          |
|at        |67   |68 |IN |long      |case           |
|long      |70   |73 |JJ |restaurant|amod           |
|last      |75   |78 |NN |long      |nsubj          |
|.         |79   |79 |.  |lunch     |punct          |
+----------+-----+---+---+----------+---------------+
```

Fig. 4. Result of word dependencies in the tweet after applying the pre-trained model “dependency_typed_conllu” in PySpark Notebook

4.3. Calculations

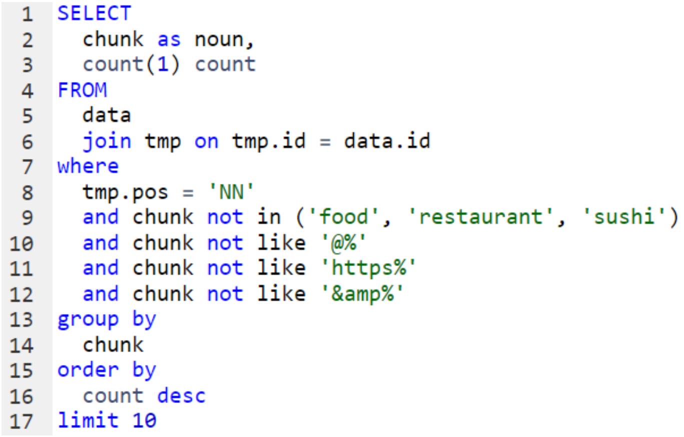
Data calculation takes place in two stages. The first stage is to calculate the “most frequently used words with the word class type ‘noun’” for a specific emotion, for example negative emotions. The second stage calculates “the dependency of the most frequently used noun words with some measure”. It is determined that measurements are those words in the text whose word class is a number and the related word of the number is a unit of measurement, for example, if the related word is “minutes”, then it can be attributed that the measurement is about time, but if the related word of the number is “EUR”, then it can be attributed that the measurement is about costs.

To calculate the “most frequently used words with word class type ‘noun’” it is necessary to:

- Count words whose word class is a noun and where, for example, the negative emotion coefficient is greater than the coefficients
- Exclude words that are not useful for analysis, such as – technical words (https, @), words that were used to filter reviews and comments about a specific field; this is necessary, because words that were used as filters for data extraction will most often appear as the most frequently used words.

Use case example of first calculation phase

The most frequently used nouns were calculated. Fig. 5 shows the Spark SQL query.



```sql
SELECT
  chunk as noun,
  count(1) count
FROM
  data
  join tmp on tmp.id = data.id
where
  tmp.pos = 'NN'
  and chunk not in ('food', 'restaurant', 'sushi')
  and chunk not like '@%'
  and chunk not like 'https%'
  and chunk not like '&amp%'
group by
  chunk
order by
  count desc
limit 10
```

Fig. 5. PySpark SQL statement to calculate the most used nouns in Notebook

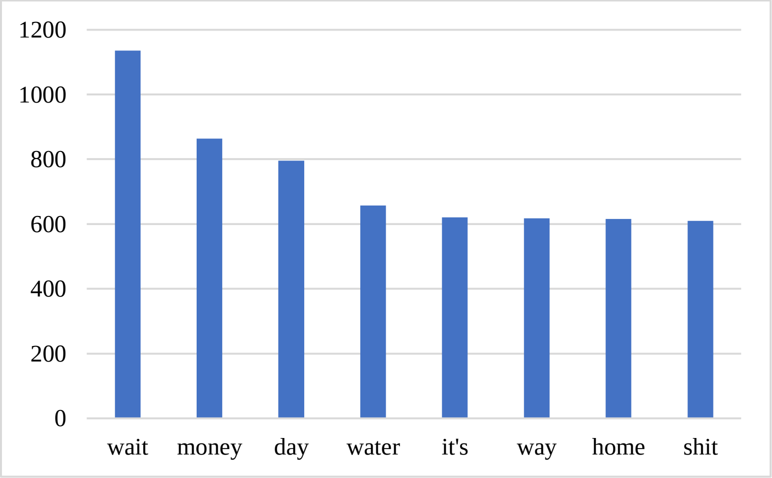


Fig. 6. Most used nouns with negative emotions

Fig. **6** shows the results obtained. From the results it can be concluded that customers often express themselves negatively if the word “time”, “money”, “day”, “shit” is in the tweet. The graph’s vertical axis represents the count of the most used words in tweets, but on the horizontal axis the most used words in tweets are represented. These words can indicate what people are most often dissatisfied with in the field of catering. For example, the word “time” could mean that people are frustrated when they have to wait a long time for food. The word “money” could indicate that the food was expensive or that the food or service was of poor quality for the price. The word “day” could be associated with the saying “waiting all day long”, which could also indicate that people are frustrated when they have to wait a long time.

In order to calculate the “the dependency of the most frequently used noun words with some measure” it is necessary to:

- Search for information about one of the most frequently used nouns
- Add a condition that the text must contain a measurement, that is, a word with the word class number, and also add a condition that defines the unit of measurement, for example “minutes”; this determines that the measurements are in the same unit of measurement and can be compared with each other
- For a specific measurement (number), calculate the average values for each of the types of emotions

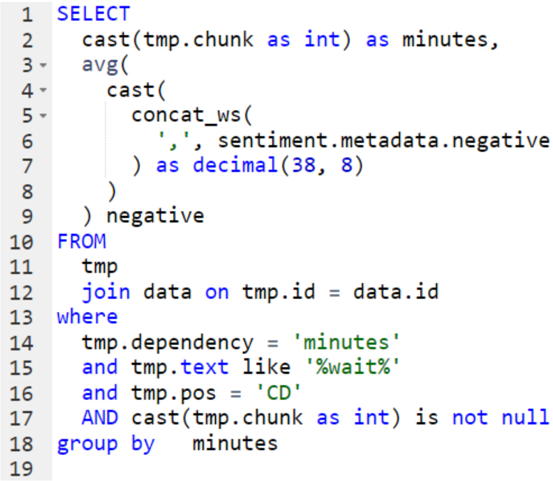
After performing these operations, the relationship between the average values of the measurements and emotions is obtained, see Table 4. Since information was only searched for a particular most frequently used word, this obtained relationship can be attributed to this most frequently used word. Table 4, column M, denotes the obtained measurement – numbers $M_1..M_n$, while columns $E_1..E_n$ denote types of emotions, for example sentimental, sarcasm, etc., for which the average values – coefficients of the corresponding elements of the CE_M table are calculated.

Table 4. Connection between most frequently used noun words and measurement

M	**F_n**	**E_n**
M_1	C_{E1M1}	C_{EnM1}
M_n	C_{E1Mn}	C_{EnMn}

Use case example of second calculation phase

Fig. 7 shows a Spark SQL query to find out the relationship between the most frequently used nouns and measurements. The pre-calculated data obtained in the “NLP processing” step of the method and stored in HDFS are used. In the given case, measurements with the unit “minutes” are searched for the noun “wait”. In the SQL query, it can be seen that words of POS type “CD” (CD – Cardinal number) – line 16 are searched for, and the word dependency with this word must be “minutes” – line 14, and the tweet containing this combination of words must have a word “wait” – line 15.



```
SELECT
  cast(tmp.chunk as int) as minutes,
  avg(
    cast(
      concat_ws(
        ',', sentiment.metadata.negative
      ) as decimal(38, 8)
    )
  ) negative
FROM
  tmp
  join data on tmp.id = data.id
where
  tmp.dependency = 'minutes'
  and tmp.text like '%wait%'
  and tmp.pos = 'CD'
  AND cast(tmp.chunk as int) is not null
group by   minutes

```

Fig. 7. PySpark SQL statement to calculate nouns with measurement in Notebook

Table 5 shows the relationship as people's negative emotions increase over time. It can be observed in the table that people's negative emotions are increasing over time. This may indicate that customers express themselves negatively, for example, if they spend a longer time in line.

Table 5. Time and negative emotion coefficient relation

Time (m)	Negative coefficient
4	0.15819366
5	0
15	0.71939789
20	0.99982475
24	0.00004389
25	0.9999994
30	0.78878643
35	1
40	1
45	0.49657143
50	1

4.4. KPI themes

Customer reviews and comments on a specific topic, which are strongly positive or negative, are important details, because they have caused the emotions of the organisation’s customers – positive or negative. These customer reviews should be used as the source data for new KPIs generation – KPI themes. This means that the most frequently used nouns in user reviews and comments determine the topics for which it is necessary to create new KPIs. The most frequently used nouns were found out in the method step “Calculations” in the calculation “most frequently used words with the word class type “noun””. Not all nouns could be applicable as KPI themes; before accepting a noun as a KPI theme, a manual review is needed.

Use case example of KPI theme determination step

The calculation of “the most frequently used words with the word class type “noun”” revealed that the most frequently used nouns with a negative coefficient are wait, money, day, water, it’s, way, home, shit. Not all of these nouns are applicable as KPI themes; at this point it would be relevant for a manual analyst input to include those nouns that are appropriate for KPI themes. For example, in some cases it would be more appropriate to use one of the synonyms of the word. In the specific case, nouns that would be suitable for KPI themes are wait, money, quality (from the word shit).

4.5. Quantitative KPI values

This step of the method involves determining the quantitative values of the KPI. In order to determine the quantitative values of the KPI, it is necessary to use the information calculated in the previous stages – “for the most frequently used nouns in connection with the measurement”.

The principle of how to automatically determine the quantitative importance of KPI is to set the goal that the value of KPI should not be greater than the average value obtained from user reviews on a specific topic, for example, waiting time. This means that the average value of the coefficients of a particular emotion using the formula

$$A = \frac{1}{n}\sum_{i=1}^{n} e_i = \frac{a_1+a_2+a_n}{n},$$

where $a_1..a_n$ is mean values of emotion type coefficients and n is the number of coefficients, must be determined. The average value obtained represents the line $Y=A$ in the graph.

In order to use the coefficient values of a specific emotion, the author assumes that a trendline must be obtained from the beginning. A trendline graph represents a trend as a curve moves through the available values. Trendline determination is necessary because, as in the example in Fig. 8, it can be seen that the values of the coefficients change rapidly and, for example, at the 24-minute mark, it might not provide a characteristic measurement for the general trend. The trendline is calculated using the values of emotion coefficients and the function $ax^2 + bx + c = y$.

In the graph, the quantitative value of the KPI is the intersection point of the straight line representing the average value of the user feedback coefficients values and the trendline of the user feedback coefficients.

Use case example of quantitative KPI value determination step

In user comments about waiting time, the average value of negative emotion coefficients is 0.662386012, so *Y = 0.662386012*, while the trendline (parabolic function) is

$$y = -\,0.000267516\,x^2 + 0.0314991x + 0.064777569.$$

Trendline (parabolic function) is obtained from the known X (time) and Y (emotion coefficients) values. In the graph (Fig. 8), the y-axis represents the coefficient of negative emotion, while the x-axis represents time (minute) values. The graph shows that the coefficient of negative emotion at the 4 and 5-minute mark is low; it is less than 0.15, but when it reaches the 15 minute mark it rises rapidly; at the 20 minute mark, it has already reached the maximum value coefficient of 1. At the 24-minute mark, it has fallen again; in this case, this result is difficult to interpret, but the general trend still remained. At the 30-minute mark, it fell slightly, but is still strongly negative at ~0.79. The graph shows that at the 45-minute mark, the negative coefficient decreases; this can probably be explained by home deliveries, which would be an acceptable value in such cases. In the graph, the horizontal line is the average value of the negative emotion coefficient values, which is 0.679062, while the dashed line is the trendline (parabolic function of x and y values).

The method determines that the KPI value should be less than the average value from user feedback on a specific topic. This means that the intersection of the trendline (parabola) and the straight line indicates the quantitative value of the KPI. In this case, it is 24 minutes.

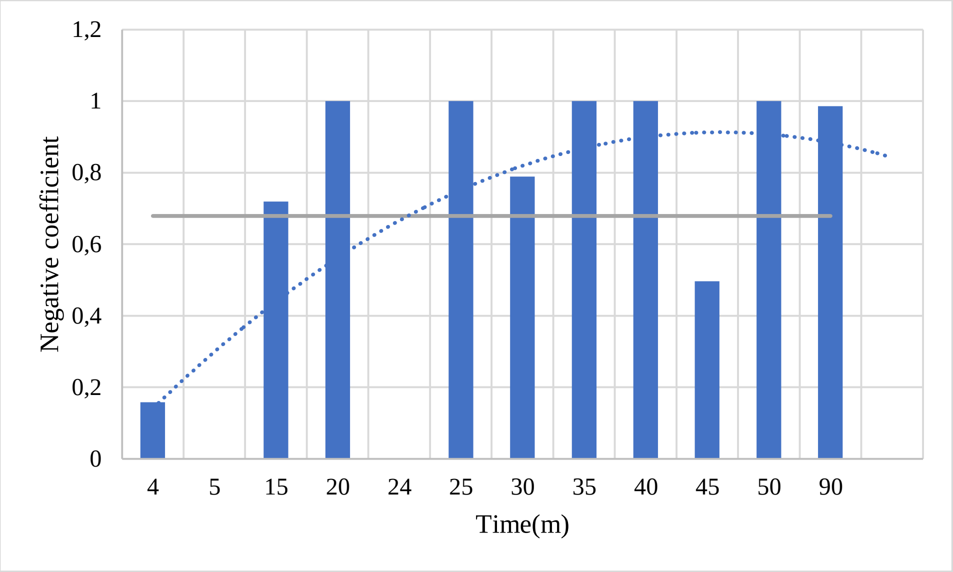


Fig. 8. Negative emotion’s coefficient trendline and average values in time

4.6. KPI conditions

In the previous steps of the method (subsection “KPI themes”), KPI themes were clarified – nouns about which customers often express emotions, for example “wait time”. To generate quantitative KPI, it is necessary to choose the appropriate KPI condition. The method stipulates that KPI conditions must be chosen according to the KPI theme. Customers usually expect that KPI themes like “price” and “wait time” are a smaller value, therefore the “less than (<)” condition should be applied, for example “price < EUR 12”. But KPI themes like “insurance compensation amount” and “available tables at a restaurant” are expected to be a higher value, therefore the “greater than (>)” condition should be applied, for example “insurance compensation amount > EUR 600”.

Table 6, NT_1 denotes a specific type of KPI theme, which can be equated as “higher” or “less” condition. This step must be done manually for each KPI theme.

Table 6. Conditions of KPI themes

KPI theme	Condition
NT_1	Higher or less (> or <)
NT_n	Higher or less (> or <)

Use case example of determination step of KPI conditions

Table **7** shows the attribution of KPI themes which were chosen for this paper’s use case. For these KPI themes only the “less” conditions were obtained.

Table 7. Use case KPI theme conditions

KPI theme	Condition
Price	less (<)
Wait time	less (<)

4.7. KPI generation

A simple quantitative KPI consists of a noun, a condition, a KPI target value, and a unit of measure. The method stipulates that the KPI themes identified in advance should be used as nouns – the most frequently used words with the word class type “noun”, Fig. **6**. The condition is applied from the appropriately defined theme for each KPI, see Table **7**. KPI target value, and a unit of measure is calculated from emotion coefficients and word dependencies in sentences, see Fig. 8. The technique for how to compile quantitative KPIs is illustrated in Fig. 9.

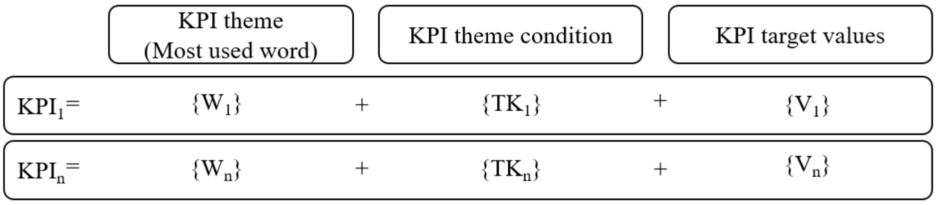

KPI theme (Most used word)
KPI theme condition
KPI target values
KPI_1= $\{W_1\}$ + $\{TK_1\}$ + $\{V_1\}$
KPI_n= $\{W_n\}$ + $\{TK_n\}$ + $\{V_n\}$

Fig. 9. Simple KPI generation from elements

Use case example of KPI generation step

In the considered example, KPI themes were clarified – "wait time", "money", "quality". The theme "wait time" was chosen to clarify the measurements. In the example, measurements related to "negative emotion" were clarified, Fig. 8. The KPI theme "wait time" is equated to the condition less than (<), see Table **7**. KPI components – the theme "wait time", the condition "<", the measurement value "24" and the measurement unit "minutes" were obtained. According to the method described, a simplified KPI is created – "wait time < 24 minutes", which can be reformulated in a more human format "Wait time should be less than 24 minutes."

4.8. Data warehouse data model improvements

The method assumes that those nouns for which measurements are made must be included in the data model of the data warehouse as attributes in fact or dimension tables. Such action should be taken so that organisations can measure new acquisitions in KPIs. Currently, the integration of these attributes into the data model is expected to be manual, but this manual operation is expected to be automated in future improvements of the method.

5. Use case

The method was applied to a case in an organisation operating in the field of catering. The operation of the method in this use case is described in the previous steps of this article to help to understand the operation of the method with real examples. This chapter describes the current situation of a selected organisation, use case technologies, data acquisition, examines the results of more experiments, and demonstrates how the data model of the organisation's data warehouse can be improved.

5.1. Current situation

There is an organisation that operates in the field of catering. The organisation has its own information system, which allows one to save information about the customer's table reservations, orders, staff, ordered dishes, menu, allergens of menu items, types of allergens. There is a small data mart in the organisation, see Fig. 10. There is a single fact table that stores the order fact, and dimension tables d_table, d_customer, d_food, and a time dimension.

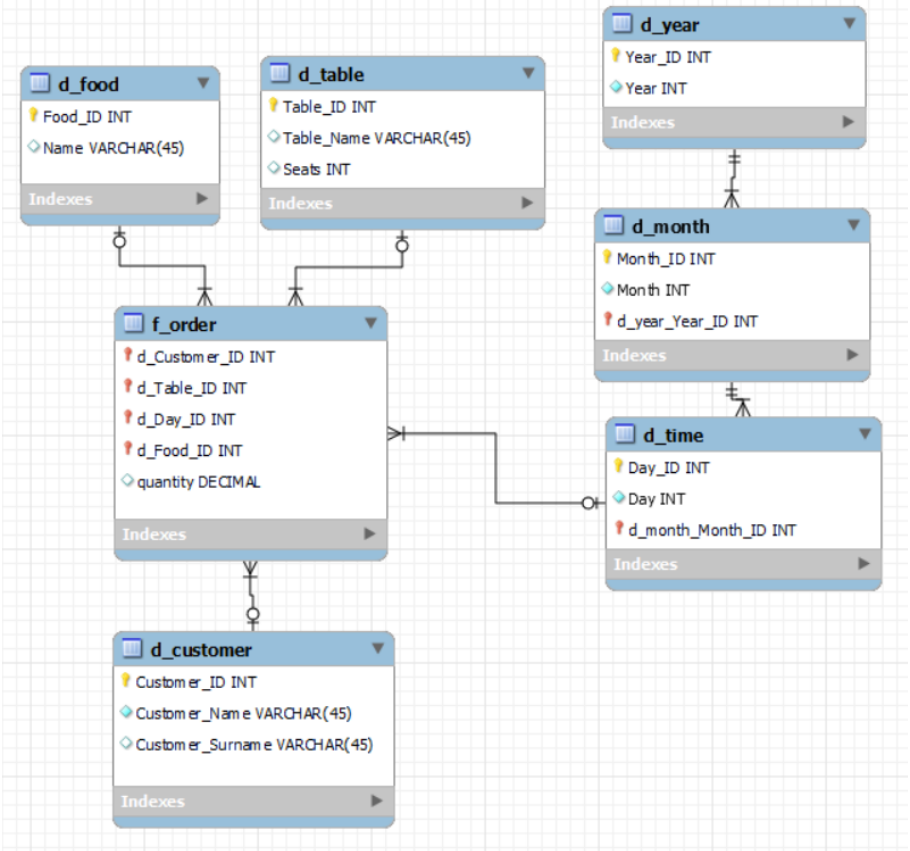


Fig. 10. Organisation’s data warehouse data model

5.2. Data gathering and technologies

For the use case example, user tweets were streamed. Tweets are freely typed text by the user, which can be up to 280 characters long. Data acquisition (streaming) from Twitter is done with Python 3.8.13 library Tweepy 3.10.0. Raw tweets are stored in the Hadoop Distributed File System (HDFS) 2.7.7. Tweet text will be processed by Apache Spark 3.0.3 using Scala 2.12.10 and with models from the SparkNLP 4.0.2 library that determine the sentiment of the text. The resulting information is stored in HDFS. Data calculations are performed using Apache Spark, and the obtained results are stored in HDFS.

5.3. Results

In the previous chapters, a use case example was described, where the relationship of one of the most frequently used words in customer reviews, “wait time”, with negative emotions in user reviews was examined, a new attribute of the KPI and data warehouse data model – “wait time” was obtained. The method assumes that such a connection can be made for other frequently used words in customer reviews. In this chapter, the results of the use case are discussed according to the same principle as described in the previous chapters.

5.3.1. The results of the relationship “Wait time – sadness coefficient”

Fig. 11 shows the connection between the words “wait time” and the emotion “sadness”. The graph shows that the emotion quotient increases over time. The average value in the user reviews of the emotion “sadness” is ~4.1, a trendline is calculated from the known values of the “dependency of the most frequently used noun words with some measure”. The crossing point of a straight line and a parabola is 24.5. In this case, the KPI theme is “wait time”. According to the technique discussed in the chapter “KPI conditions”, the KPI theme “wait time” is equated to less (<). A simplified KPI “wait time < 24.5” is obtained according to the description of the method.

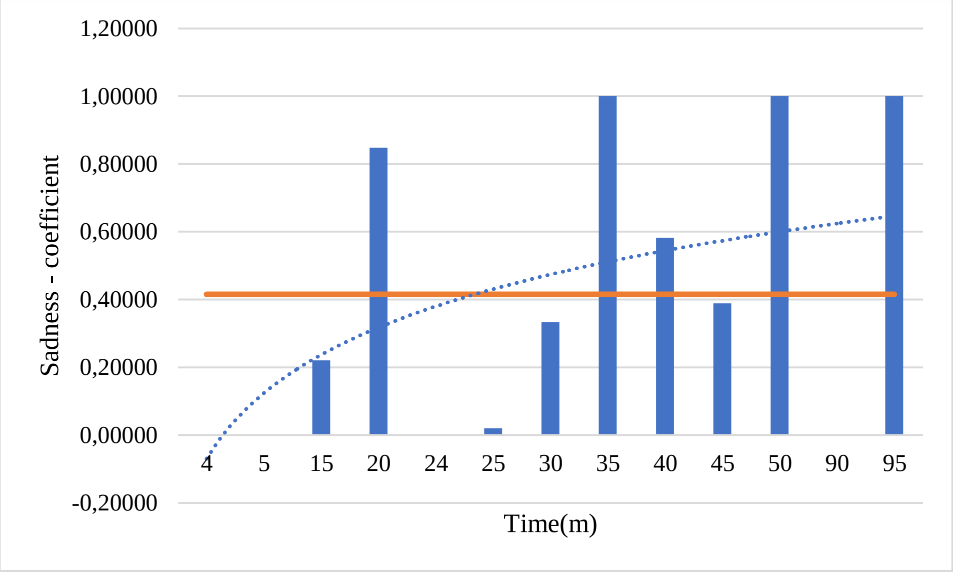


Fig. 11. Sadness emotion’s coefficient trendline and average values in time

5.3.2. The results of the relationship “Wait time – joy coefficient”

Fig. 12 shows the connection between the word “wait time” and the emotion “joy”. A simplified KPI “wait time <25” is obtained according to that which is specified in the method.

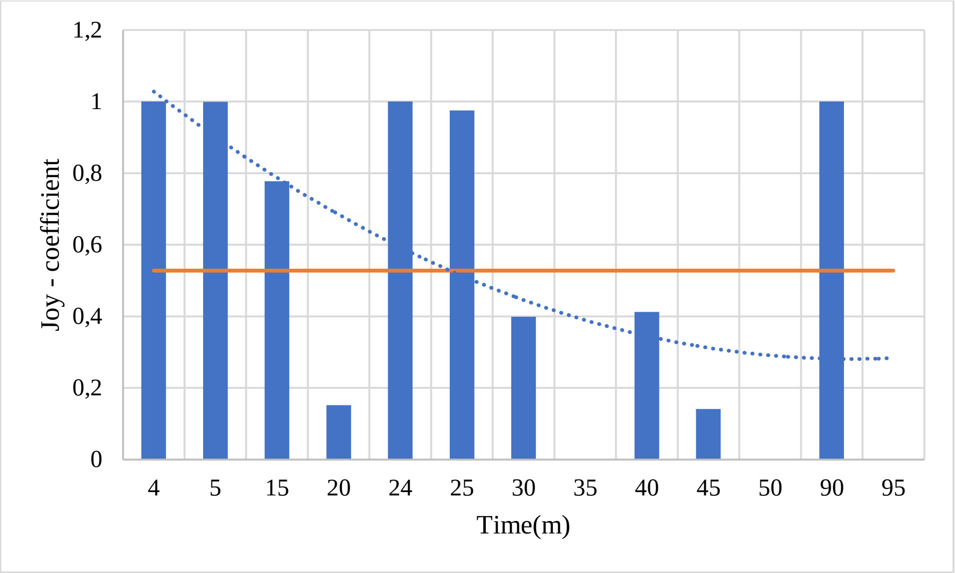


Fig. 12. Joy emotion's coefficient trendline and average values in time

5.3.3. The results of the relationship "Price – negative coefficient"

Fig. 13 shows the connection between the word "price" and the emotion "negative". A simplified KPI "price < EUR 12" is obtained according to that which is specified in the method.

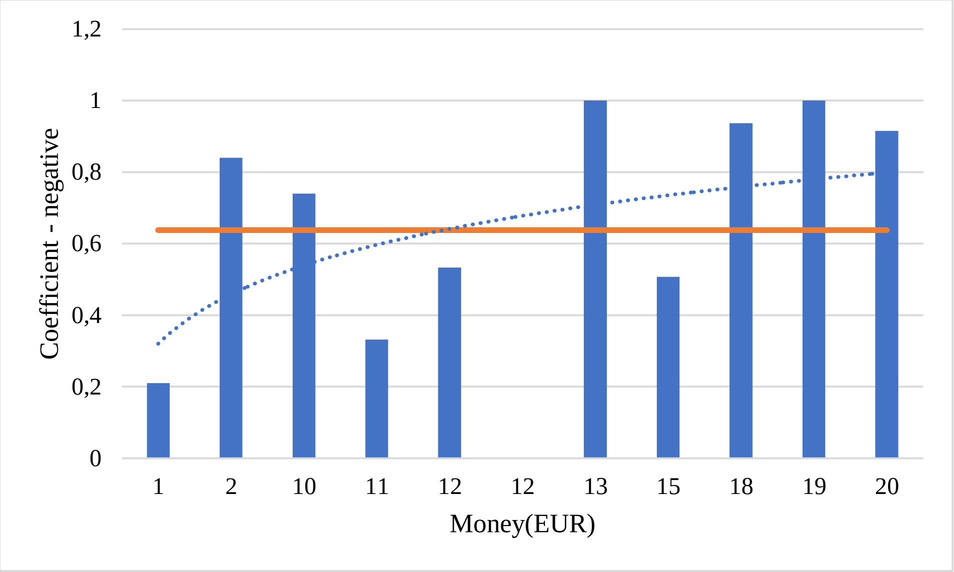


Fig. 13. Negative emotion's coefficient average value and negative emotion's coefficient and price trendline

5.3.4. The results of the relationship “Price – surprise coefficient”

Fig. 14 shows the surprise factor and emotions. According to that which is specified in the method, a simplified KPI “price < EUR 12” is obtained.

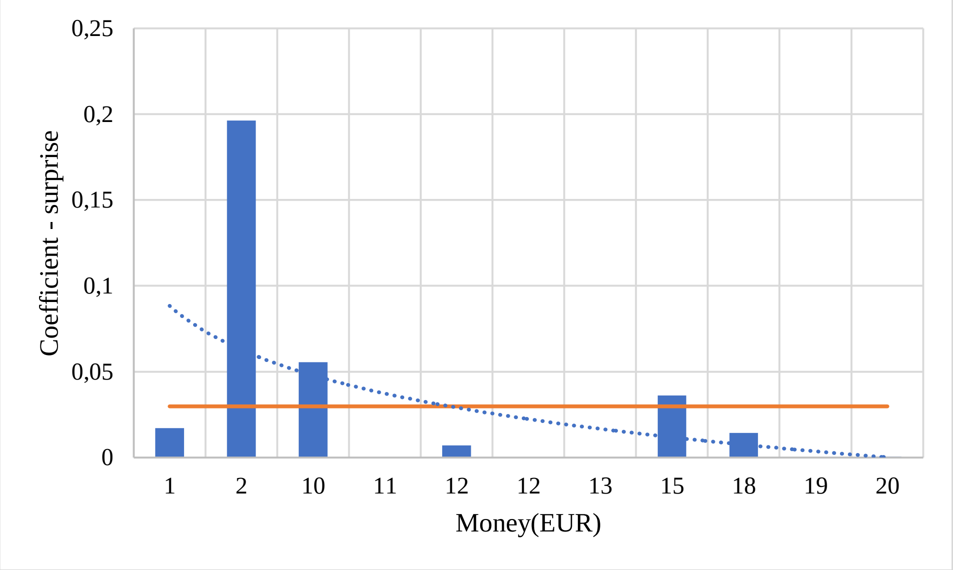


Fig. 14. Surprise emotion’s coefficient average value and surprise emotion’s coefficient and price trendline

5.4. Average values from results

In the previous section we can see that for one theme more than one KPI target was obtained. This is a normal situation because one theme could have more than one emotion type and each emotion type could have different results. In such a situation, the average value needs to be calculated from target values for a particular theme. Table **8** shows the theme’s target values and its average value. KPI with the theme “wait time” target value should be 24.75.

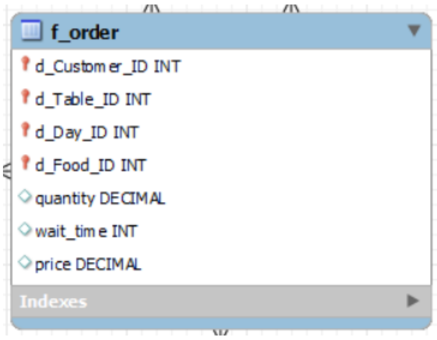
Table 8. Theme “wait time” results

KPI theme	Condition	KPI target value
wait time	<	24.5
wait time	<	25
Average		**24.75**

5.5. KPI and DW data model improvements

Information requirements for the DW data model are obtained from the KPIs according to the methods specified in the method. The KPI “wait time < 24.75” results in a simple DW data model requirement to represent the “wait time” attribute, while the KPI “price < 12” results in a simple DW data model requirement representing the ‘price’ attribute.

Fig. 15 shows how the fact table of the data warehouse data model has been augmented with two attributes “wait_time” and “price”. In the particular case, two attributes were found, but new attributes can be found as new information is read.



f_order

- d_Customer_ID INT
- d_Table_ID INT
- d_Day_ID INT
- d_Food_ID INT
- quantity DECIMAL
- wait_time INT
- price DECIMAL

Indexes

Fig. 15. Data warehouse fact table “f_order” after new attributes

6. Conclusions

This article describes a new method for improving the data model of a data warehouse using customer feedback – unstructured data. The method specifies that user reviews are processed by a natural language processing tool. Information is obtained about word dependencies in sentences and emotions (sentiment analysis) in customer reviews. The method includes multiple calculations that are made to obtain essential elements for KPIs and their quantitative value generation. Attributes from the generated KPIs must be stored in the data warehouse, in order for the organisation to monitor the performance of the indicators. Storing attributes in a data warehouse allows organisations to monitor the development of the organisation and improve its operation.

This method allows one to find new data warehouse data model attributes using customer reviews as the source data. The method does not allow one to generate a new data warehouse data model from scratch, but it is still a very useful and powerful way to enhance the existing data warehouse data model with new attributes.

In order to apply the described method, organisations must have customer reviews available about the field in which they are operating in. Customer reviews may not be from the organisation’s clients, but the reviews must be related to the field in which they operate in; for the use case in this article for example, public Twitter data was used. In order to test and validate the method, other field data (insurance, accommodation, shopping) was loaded and analysed. In all cases it was observed that in order to create quantitative KPIs, the data to be processed must contain feedback that characterises a measurement.

This method is suitable for organisations with an existing data warehouse. The organisation must operate in a field where customer feedback is available, for example financial services, insurance, e-commerce, etc., because the method processes customer feedback. The method might not be applicable in the case where the organisation does not monitor the performance of the indicators.

The article described a simple use case of the method's application in the field of catering. Such use case was chosen to be able to describe the operation of the method in a simpler way. For this use case, real Twitter data was used as the data source. The use case resulted in two new attributes of the data warehouse data model, five new KPIs and their target values.

This is the method's first version, and it could be improved in several areas:

- Currently method steps must be executed manually, similarly to this article's use case. An improvement of the method would be making the execution automated
- Graphical user interface would help users interact and configure method execution
- Qualitative/Soft KPIs determination. Currently the method can only be used for Quantitative/Hard KPI determination, but it is also possible to obtain new KPIs and improve the data model of the data warehouse from Qualitative/Soft KPIs
- Improve the method's step "Data warehouse data model improvements" to integrate newly acquired data warehouse data attributes in the existing data warehouse data model
- Develop functionality to store and maintain obtained method results in database
- Calculation steps of the method could be improved by also taking into account sentiment neutral emotions, not only positive and negative
- Currently the method calculates synonyms separately; the method could be improved by calculating synonyms together

References

Baviskar, D., Ahirrao, S., Kotecha, K. (2021). Multi-layout unstructured invoice documents dataset: A dataset for template-free invoice processing and its evaluation using AI approaches. *IEEE Access*, **9**, 101494-101512.

Benkhaled, H. N., Berrabah, D. (2019). Data Quality Management For Data Warehouse Systems: State Of The Art. JERI.

Bimonte, S., Antonelli, L., Rizzi, S. (2021). Requirements-driven data warehouse design based on enhanced pivot tables. *Requirements Engineering*, **26**, 43-65.

Bouaziz, S., Nabli, A., Gargouri, F. (2019). Design a data warehouse schema from document-oriented database. *Procedia Computer Science*, **159**, 221-230.

Briggs, B., Hodgetts, C. (2017). Tech trends 2017: An overview. In *Wall Street Journal*.

Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., Kurzweil, R. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Chowdhary, K. R., Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.

Danaher, P. J., Smith, M. S., Ranasinghe, K., Danaher, T. S. (2015). Where, when, and how long: Factors that influence the redemption of mobile phone coupons. *Journal of Marketing Research*, **52**(5), 710-725.

Domínguez, E., Pérez, B., Rubio, Á. L., Zapata, M. A. (2019). A taxonomy for key performance indicators management. *Computer Standards & Interfaces*, **64**, 24-40.

Doshi, R. D., Sidpara, C. B., Khimani, K. U. (2016). Automatic Metadata Harvesting from Digital Content Using NLP. In *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems:* Volume 1 (pp. 479-485). Springer International Publishing.

Garg, R., Kiwelekar, A. W., Netak, L. D., Bhate, S. S. (2021). Potential use-cases of natural language processing for a logistics organization. In *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI*, Volume 2 (pp. 157-191). Cham: Springer International Publishing.

Howatson, A. (2016). How to unlock the power of unstructured data. *Marketing Tech News*.

Inmon, W. H. (2005). *Building the Data Warehouse*. John Wiley & Sons.

Kaldeich, C., Sá, J. O. E. (2004). Data warehouse methodology: A process driven approach. In *Advanced Information Systems Engineering: 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004. Proceedings 16* (pp. 536-549). Springer Berlin Heidelberg.

Kozmina, N., Niedrite, L., Zemnickis, J. (2017). Gathering Formalized Information Requirements of a Data Warehouse. In *ICEIS* (1) (pp. 217-224).

List, B., Bruckner, R. M., Machaczek, K., Schiefer, J. (2002). A comparison of data warehouse development methodologies case study of the process warehouse. In *Database and Expert Systems Applications: 13th International Conference, DEXA 2002 Aix-en-Provence, France, 2-6 September 2002 Proceedings 13* (pp. 203-215). Springer Berlin Heidelberg.

Navinchandran, M., Sharp, M. E., Brundage, M. P., Sexton, T. B. (2021). Discovering critical KPI factors from natural language in maintenance work orders. *Journal of Intelligent Manufacturing*, 1-19.

Niedritis, A., Niedrite, L., Kozmina, N. (2011). Performance measurement framework with formal indicator definitions. In *Perspectives in Business Informatics Research: 10th International Conference, BIR 2011, Riga, Latvia, 6-8 October 2011. Proceedings 10* (pp. 44-58). Springer Berlin Heidelberg.

Parmenter, D. (2015). *Key performance indicators: developing, implementing, and using winning KPIs*. John Wiley & Sons.

Pejić Bach, M., Krstić, Ž., Seljan, S., Turulja, L. (2019). Text mining for big data analysis in financial sector: A literature review. *Sustainability*, **11**(5), 1277.

Pernici, B., Francalanci, C., Geronazzo, A., Lucia, P., Stefano, R., Leonardo, R., Todor, I. (2018). Relating big data business and technical performance indicators. In *Proceedings ITAIS 2018: XV Conference of the Italian Chapter of AIS* (pp. 1-12).

Prakash, D., Prakash, N. (2019). A multifactor approach for elicitation of information requirements of data warehouses. *Requirements Engineering*, **24**, 103-117.

Rizkallah, J. (2017). The big (unstructured) data problem. Forbes. Retrieved on 5 September 2017.

Wang, J. (Ed.). (2008). *Data warehousing and mining: Concepts, methodologies, tools, and applications* (Vol. **3**). IGI Global.

Wang, J., Xu, C., Zhang, J., Zhong, R. (2022). Big data analytics for intelligent manufacturing systems: A review. *Journal of Manufacturing Systems*, **62**, pp. 738-752.

Winter, R., Strauch, B. (2003). A method for demand-driven information requirements analysis in data warehousing projects. In *The 36th Annual Hawaii International Conference on System Sciences*, 2003. Proceedings of the IEEE (pp. 9)..

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E. V., Batista-Navarro, R. T. (2021). Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys (CSUR)*, **54**(3), pp. 1-41.

Received May 4, 2023, revised August 7, 2023, accepted September 9, 2023