World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:12, No:4, 2018

Evaluating 8D Reports Using Text-Mining

Benjamin Kuester, Bjoern Eilert, Malte Stonis, Ludger Overmeyer

***Abstract*—**Increasing quality requirements make reliable and effective quality management indispensable. This includes the complaint handling in which the 8D method is widely used. The 8D report as a written documentation of the 8D method is one of the key quality documents as it internally secures the quality standards and acts as a communication medium to the customer. In practice, however, the 8D report is mostly faulty and of poor quality. There is no quality control of 8D reports today. This paper describes the use of natural language processing for the automated evaluation of 8D reports. Based on semantic analysis and text-mining algorithms the presented system is able to uncover content and formal quality deficiencies and thus increases the quality of the complaint processing in the long term.

***Keywords*—**8D report, complaint management, evaluation system, text-mining.

International Science Index, Computer and Information Engineering Vol:12, No:4, 2018 waset.org/Publication/10008745

I. Introduction

EIGHT disciplines ensure the proper complaint handling according to the 8D method. It is triggered when an internal or external customer complains about a product. In the event of a legitimate complaint because the supplier delivered a defective product, the supplier is obligated to eliminate the defect through the structured procedure of the 8D method and following that to send the 8D report to the customer. The report therefore acts as a proof for the customer and as a customer loyalty instrument. However, it is far more important for internal process improvement [1]. According to the concept of Total Quality Management (TQM), the 8D method is an essential element in order to continuously achieve product quality and process stability with the zero-defect philosophy [2]. Unfortunately, the reports resulting from the 8D method have too many errors themselves to create a profitable benefit. The errors are both, content as well as formal, and arise to a large extent because of the main source of error – the human. In line with an analysis of the sources of error, the following core causes for bad 8D reports were identified:

1) Lack of appreciation among decision-makers (no added value)
2) Lack of acceptance by employees (annoying secondary activity)
3) Lack of methodological knowledge
4) Lack of process knowledge
5) Lack of temporal free spaces
6) Lack of coordination between quality, design, production

B. Kuester is with the Institut fuer Integrierte Produktion Hannover gGmbH, 30149 Hannover (corresponding author, e-mail: kuester@iph-hannover.de).

B. Eilert, M. Stonis, and L. Overmeyer are with the Institut fuer Integrierte Produktion Hannover gGmbH, 30149 Hannover (e-mail: eilert@iph-hannover.de, stonis@iph-hannover.de, overmeyer@iph-hannover.de).

A prerequisite for the lasting effect of proper handling of complaints is the conscientious execution of the company and the conviction of the company to convey its own employees. In the industrial environment driven by enormous cost pressure, the quality demands on products increase as much as the number of units. Qualitatively better products decide the success of a company as well as solvency or insolvency. No company can afford to make too many mistakes - and yet of course they happen. The key question is how to deal with these errors. The ambition, especially of manufacturing companies, should be to learn from the mistakes and never let them happen again. In a recent survey, it was determined that over 70% viewed the 8D method as a helpful and appropriate instrument to handle complaints in a targeted manner [3]. However, the quality does not meet expectations and leads to ambiguities and dissatisfaction.

II. Automated Evaluation of 8D Reports

In order to remedy this inconvenience and to promote continuous product and process improvement with the aid of the 8D method, a software system has been developed that evaluates completely processed 8D reports with regard to formal and content quality. If the 8D method is triggered internally due to a customer complaint, the method first runs through the usual process. All eight disciplines are processed by the competence team. Fig. 1 shows all eight steps within the 8D method.

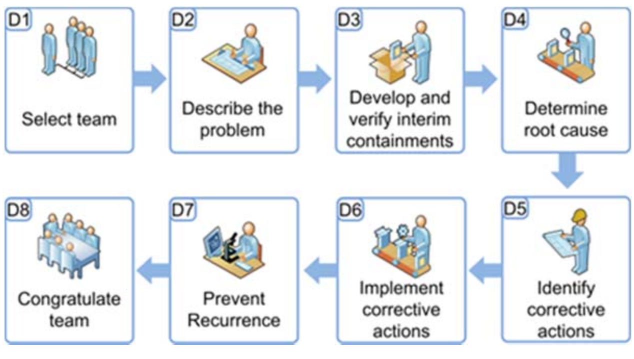

Fig. 1 Eight disciplines of the 8D method

The 8D report is documented in parallel to the problem solution and is handed over to a responsible person or rather quality manager at the end of the process. Depending on the company size, a thorough review of the finalized 8D reports is difficult to perform by one or more people. Many companies do not carry out any checks or only isolated samples. This problem is resolved with an automated system. The developed system is capable of evaluating every 8D report in a uniform and sound manner. The evaluation is carried out on the basis of quality criteria (e. g. intelligibility, quantification, solidity), which can be defined and weighted in advance within the system. Each criterion is evaluated individually and independently of the system. The results of each criterion are aggregated within each discipline and summarized for the entire report to a total score (0 - 5 points). On this basis, it is possible to identify high error accumulations and to investigate them in a targeted manner. If the overall rating is below a tolerance limit, the quality commissioner can refuse the release and the forwarding of the report to the customer, and then arrange the report to be reprocessed and corrected.

All checks carried out with the system are recorded and processed in an integrated statistics module. This second important function allows to categorize and understand the history of evaluations within a certain period or for a particular product. Process-related weaknesses can be detected. As a result sound countermeasures can be taken which will sustainably improve the complaints process. For example, targeted training for employees can be carried out.

III. System Design

A. Overview

The system is largely based on new findings and methods in the field of computer linguistics. Computer linguistics has its origin in the ‘50s of the 20th century. Computer linguistics deals with the automatic processing of natural language. This makes it a sub-area of artificial intelligence and, at the same time, an interface between computer science and linguistics [4]. Thus, appropriate programming techniques and languages as well as the construction of efficient algorithms and memory technologies are used. From the linguistic sciences, the concepts for the description of linguistic sounds (phonetics), word formation (morphology), sentence structure (syntax), meaning (semantics) and usage (pragmatics) are used. Computer linguistics also uses statistics and logic. The fields of application of computer linguistics are speech recognition and synthesis, automatic translation into other languages and information extraction from texts. Compared to a computer, people can easily and intuitively understand, apply and process natural language information. Natural language includes features such as ambiguity, competing semantically equivalent expressive possibilities, and vagueness that make machine understanding of natural language more difficult [5].

B. System Structure

While technically the NLP or rather text-mining is the most important element for the content evaluation, it is only a part which is integrated in the entire system. The structure of the complete system is shown in Fig. 2. The interface between the text-mining algorithms and the company model is the metric system. Metrics define which information must be processed and exchanged between company knowledge and 8D report. Each metric interprets the results of a specific quality criterion in a mathematical index. In the end, the metric system aggregates the indexes for formal and content criteria to a final quality evaluation. Besides the decimal number, the evaluation is shown with one to five stars for illustrative and intuitive reasons.

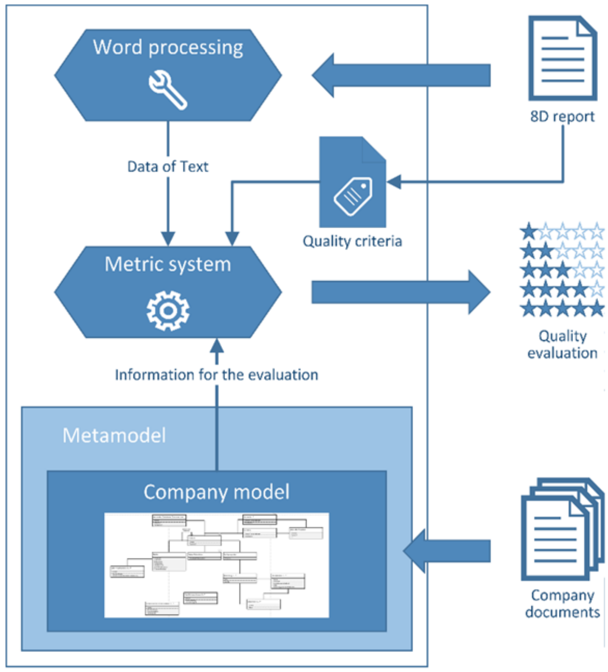

Fig. 2 Structure of the evaluation system

C. Process of Evaluation

Within the developed system, the evaluation of 8D reports is divided up in four steps (Fig. 3).

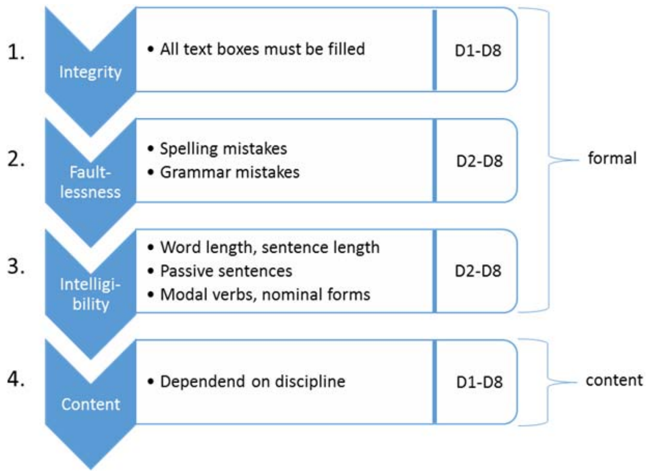

Fig. 3 Process of evaluating 8D reports in four steps

First, a simple algorithm is used to check whether all relevant fields of the 8D report are filled out. Only in case of fully completed formulas the next step will be continued. The completeness is the basic prerequisite for a high-quality 8D report and the evaluation. The second stage deals with the determination of spelling and grammar errors. The formal evaluation is completed by the assessment of the intelligibility and readability. In contrast to the formal evaluation, which is carried out for each discipline the same way, the assessment of the content takes place in the fourth stage, depending on individual disciplines.

D. Formal Evaluation with Spell Check

As a spelling check, software-supported methods are used for the detection of spelling mistakes in electronically available natural language texts. Today, the spelling check is not only used in word processing programs (e.g. Microsoft Word, Mozilla Thunderbird), but also in many other programs (e.g. Microsoft Visual Studio).

Today's spelling-checks use morphological processes to perform the spelling check as quickly as possible [6]. The input text is analyzed word by word. It tries to find the word within a dictionary or to compose it by means of rules from words in the dictionary. If the word is unknown, it is marked as a spelling error and possible correction suggestions are specified. Besides that the Levenshtein distance (1) is used for the system. This provides a measure of the degree of similarity/dissimilarity of two character strings. The idea behind this is that the similarity is based on the number of operations (insert, delete and replace) needed to translate word1 into word2 [7].

$$lev_{a,b}(x,y) = \begin{cases} \max(x,y) \\ \min \begin{cases} lev_{a,b}(x-1,y)+1 \\ lev_{a,b}(x,y-1)+1 \\ lev_{a,b}(x-1,y-1)+1_{(a_x \neq b_y)} \end{cases} \end{cases} \quad (1)$$

Due to the variability, the dictionary size and the actuality, the spelling check was implemented with the software Hunspell.

Hunspell is a free software for spelling check based on the work of an open source morphological analyzer [8]. It was originally intended only for the Hungarian language, hence the name Hunspell. Hunspell is a development of MySpell. Hunspell uses the UTF-8 encoding in comparison to the 8-bit ASCII encoding of MySpell to encode all characters. Hunspell is ideal for languages with rich morphology and complex word formation. It is used among others in OpenOffice, Google Chrome, Mozilla Firefox, Thunderbird and SeaMonkey. For processing natural language texts, Hunspell requires two files: a dictionary file (*.dic) and an affix file (*.aff). The dictionary file contains a simple compilation of words and parts of the word. Each word or word part is arranged in a separate line. Some words are supplemented with markers that determine and limit the reciprocal combinability of these elements.

The Affix file initially contains various specific properties of the dictionary (for example, character encoding, frequent character substitution). Mainly, the affix file contains a list of affixes (prefixes and suffixes) corresponding to the individual masks from the word list. This list also shows which affixes can connect to which words to form a new word. Each affix is described by a set of rules that clearly define its behavior.

E. Content Evaluation with POS Tagging

Part-Of-Speech Tagging (POS tagging) is a precursor for the further syntactic and semantic processing of natural-language texts. It plays an important role for the assessment of the intelligibility as well as legibility and serves as a basis for the content and semantic evaluation. In POS tagging, word types are determined from natural language texts. As a rule, POS taggers work in three steps: tokenize, determine the POS tag for each token, and select a POS tag per token using a language model. For tokenization, natural-language texts are divided into their individual parts, so-called tokens. The natural language text is divided with the identification of spaces or punctuations. In the following example the tokenized original text contains eight tokens.

Original text:
The screw is broken.

After tokenization:
| The | screw | is | broken | . |

The POS tagger analyzes each token and assigns it to a POS tag. The POS tag contains information about the word type of a token and assigns it the corresponding word form (e.g. noun, verb, adjective, etc.). Depending on the word type, the POS tag also contains additional semantic, syntactic or morphological information [4].

In the following example, the original text is tokenized and assigned to POS tags. The tags are separated from the words by a slash (/) and appended directly to the word.

Original text:
The screw is broken.

After POS tagging:
The/ART screw/NN is/VAFIN broken/ADJD ./$

Five requirements for a POS tagger were defined [9]. The POS tagger should be robust and can handle any input (unknown words, special characters). It must be efficient and process the input quickly. It should work as error-free as possible (error rate <5%). Furthermore, the POS tagger should be customizable and reusable. It should therefore be adapted to specific requirements of an input and be easily applicable to new tasks.

Current POS taggers are able to completely decompose sentences syntactically, taking into account the linguistic relationships of a token with the other tokens occurring in the set. Thus, possible ambiguities, given by the complexity of natural language, can be resolved. The multiple possibilities are the greatest problem regarding complete language comprehension [5]. Disadvantages of POS tags are the language dependency and corresponding availability, robustness with regard to unknown words and grammatically incorrect sentences.

Two POS tagging methods exist – rule-based and stochastic. Rule-based POS taggers are based on lexica and grammar rules, these are the oldest approaches [10], [11]. The rule-based POS tagger assigns a token the day that was most frequently annotated in a training kit. It takes into account that a particular tag must occur in a particular context (e.g. no relative pronoun at the beginning of the sentence). The rules for a rule-based POS tagger can be set up by hand and already few rules deliver results. However, in order to obtain good results, a great expenditure is required. Further disadvantages of the rule-based POS tagger are that used lexicons are never complete and the rule used is often complex and extensive. Furthermore, the rules cannot be used in other languages, since the rules are always corpus-specific and language-specific.

Stochastic POS tagger is based on stochastics and statistics [12], [13]. Basis for stochastic POS tagger is a dictionary that contains all possible, correct tags for a word and the probability of occurrence of any possible combination. The stochastic POS tagger assigns the most probable tag to an ambiguous token, taking the sentence structure into account. The main advantage of the rule-based POS tagger is that the stochastic POS tagger can easily be adapted to new languages and corpora by re-training. Disadvantages are that the stochastic POS tagger is strongly corpus-dependent and difficult to implement.

Due to its best results for German language, the Stanford Part-Of-Speech Tagger was used for the system [14]. The Stanford POS Tagger was developed 2003 as a component of the Stanford CoreNLP [13]. The Stanford CoreNLP is a Java application that is subject to the GNU General Public License v3 (GPL). It provides various functions for the analysis of written language. In addition, the Stanford CoreNLP provides its own interface for application programming. This makes it possible to execute individual functions of the Stanford CoreNLP (POS tagger, parser, segmenter) independently of each other. For the Stanford POS Tagger, different models are available in various languages (English, Chinese, French, etc.).

By adapting the Stanford POS tagger, important criteria for the intelligibility and legibility of natural language can be checked. The system finds in the 8D reports, for example, long sentences and words as well as accumulations of passive sentences, noun words and modal verbs. These grammatical cases make it difficult to understand a text so that an accumulation of these constructions would lead to a poorer evaluation.

The Flesch-Reading-Ease (FRE) is used for evaluating the readability of a text [15]. Adapted to the respective language, it is a simple numerical method to get information for the difficulty of a text. The formula takes into account the average sentence length (ASL) and the average word length (ASW). For the German language, the formula is as:

$$FRE_{ger} = 206{,}835 - (1{,}015 * ASL) - (84{,}6 * ASW) \tag{2}$$

In order for the formal assessment to be more precise and accurate, additional criteria are incorporated into the evaluation in the form of developed metrics. For example, the following metric (MFW) evaluates the number of filler words.

$$MFW = 1 - \frac{N_{SFW}}{N_S} \tag{3}$$

N_{SFW} : *Amount sentences with more than two filler words*
N_S : *Total amount sentences*

Too many filler leads to a poorer evaluation of the intelligibility. Further metrics in a similar form exist, for example for verbs in the perfect, modal verbs, and nominal forms.

The POS tagger also provides the foundation for the content evaluation since it returns all the words in the 8D report to the word stem, thereby enabling the semantic analysis.

Since a content evaluation always requires a reference knowledge in order to match the content to be verified with confirmed information, a meta model is laid to the system. The simplified model is shown in Fig. 4.

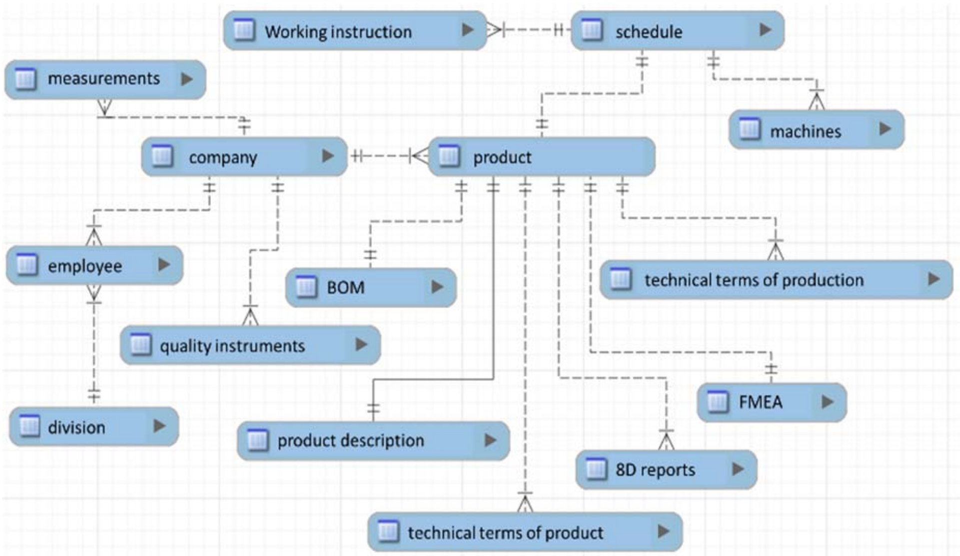

Fig. 4 Meta model as basis for content evaluation

This meta-model incorporates the knowledge necessary for the evaluation. Before use, this model is uniquely filled with company knowledge in the form of defined company documents. For this purpose, the documents are presented in a structured format and certain keywords or sentences are trained as such. The documents are stored in a cluster, which allows the assignment to specific products (e.g. bill of materials), production systems (e.g. work center description) or general information (e.g. method collection).

Depending on the product being claimed, determined company documents can be accessed specifically using the unique product identification number stated in the 8D report. Information which is not relevant to the current 8D report will not interfere the evaluation and reduce the run-time.

The conceptualities that appear in the 8D report are matched with the company's knowledge. The semantic relationship is stored in the company model or rather POS tag model. For example, if certain noun verb constellations from the corporate documents appear in the 8D report, they can be identified and evaluated accordingly.

The content evaluation is similar to a search query, e.g. with Google. The quality of the semantic search depends on two essential factors. The consideration of synonyms in the semantic search is important for the content evaluation. In the background, knowledge all known synonyms of a term are stored for this purpose. If one of these terms is used in an 8D report, all related synonyms are included in the search query. Thus, it is possible to find the term "chassis" in those documents where the specification with the synonym "running gear" is held.

An example for a metric for the content which evaluates the general context goes:

$$M_{GC}(N_{PVS}) = \begin{cases} 0, N_{PVS} < X_{PV,min} \\ \frac{N_{PVS} - X_{PV,min}}{X_{PV,opt} - X_{PV,min}}, X_{PV,min} \le N_{PVS} \le X_{PV,opt} \\ 1, N_{PVS} > X_{PV,opt} \end{cases} \quad (4)$$

$$N_{PVS} = \frac{N_{PV}}{N_S}$$

where, N_{PV}*: Amount product vocabulary (*$N_{PV} \in L_{PV}$*),* N_S*: Total amount sentences,* N_{PVS}*: Amount product vocabulary per sentence,* $X_{PV,min}$*: Minimum for* T_{PV}*,* $X_{PV,opt}$*: Optimum for* T_{PV}*.*

The distinction of homonyms (e.g. nut (eating) versus nut (screw)) in the search results increases the quality of the content evaluation. The search results found in the context of a disambiguation are automatically removed. The text-mining algorithm can recognize the context of a document and thus conclude the correct or incorrect assignment of the topic area. If, in this case, the context of the 8D report in which the search query was found is classified as a correct search result, it means a good evaluation. Conversely, documents with a wrong context compared with the stored knowledge for e.g. a product is evaluated with a bad rating.

IV. Conclusion and Outlook

With the presented approach, 8D reports can be adequately checked and evaluated. It supports the zero-error philosophy of TQM and replaces the laborious manual check process of 8D reports in an efficient form. The person responsible for the complaint process receives reliable and well-founded assessments from each 8D report before the document leaves the company. In case of a bad evaluation, the quality manager can investigate the report, intervene and demand a re-processing of the 8D method. By storing all evaluation results in an integrated statistics module, the development of the quality of 8D reports can be followed. Based on this data, targeted strategic measures can be taken to improve the quality at the relevant points in the process, the products or the employees.

As the system is equipped with open interfaces, it can be used independently of the company's infrastructure. The initial software demonstrator was designed and implemented for German language. It can easily be transferred to other languages, such as English. Besides the German language, the spelling library Hunspell and the POS tagger provide, of course, adaptions for other languages.

Developments in the area of computer linguistics and text-mining allow automatic content quality control as the software demonstrator for the evaluation of the 8D reports shows.

Due to the fact that the product quality meets ever more stringent requirements which have to be documented by means of quality documents, it is only sensible to expand an automatic check on further quality documents (e.g. audit plans, proof documents).

This paper focuses on the functionality and design of the system. For further work the practicality and accuracy of the system will be investigated. Therefore, the evaluation results of the system will be compared to evaluation results from experts.

Acknowledgment

The work of Benjamin Küster within the IGF project 18447 N of the Association for the Promotion of Applied Information Science (GFaI) was funded via the German Federation of Industrial Research Associations (AiF) in the programme of Industrial Collective Research (IGF) by the Federal Ministry for Economic Affairs and Energy (BMWi) based on a decision of the German Bundestag.

References

[1] Behrens, B.-A.; Wilde, I.; Hoffmann, M.: Complaint management using the extended 8D-method along the automotive supply chain. In: Production Engineering, vol. 1 (2007), no. 1, pp. 91-95.
[2] Appelfeller, W.; Buchholz, W.: Supplier Relationship Management. 2. ed., Gabler Verlag, Wiesbaden 2011.
[3] Küster, B.; Eilert, B.; Overmeyer, L.: Automated Quality Evaluation of 8D Reports in Context of Complaint Processing. In: Proceedings of Symposium on Automated Systems and Technologies, vol. 3. (2016), pp. 77-80.
[4] Carstensen, K.-U.; Ebert, C.; Ebert, C., Jekat, S.; Langer, H.; Klabunde, R.: Computerlinguistik und Sprachtechnologie. 3. ed., Spektrum Akademischer Verlag, Heidelberg 2010.
[5] Pellegrini, T.; Blumenauer, A.: Semantic Web. Springer Verlag, Berlin

2006.
[6] Pirinen, T. A.; Lindén K.: Creating and Weighting Hunspell Dictionaries as Finite-State Automata. In: Investigationes Linguisticae, vol. 21 (2010), pp. 1-16.
[7] Yujian, L.; Bo, L.: A Normalized Levenshtein Distance Metric. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29 (2007), no. 6, pp. 1091-1095.
[8] Németh, L.; Halácsy, P.; Kornai, A.; Trón, V.: Open source morphological analyzer. https://catalog.ldc.upenn.edu/docs/LDC2008T01/acta04.pdf. Last access: 2017-05-30.
[9] Cutting, D.; Kupiec, J.; Pederson, J.; Sibun, P.: A Practical Part-of-Speech Tagger. In: Proceedings of the third conference on applied natural language processing, vol. 3. (1992) pp. 133-140.
[10] Brill, E.: A simple rule-based part of speech tagger. In: Proceedings of the Third Conference on Applied Computational Linguistics, vol. 3 (1992), pp. 112-116.
[11] Brill, E.: A report of recent progress in transformation-based error-driven learning. In: Proceedings of the Workshop on Human Language Technology, 1992, pp. 256-261.
[12] Brants, T.: TnT- A Statistical Part-of-Speech Tagger. In: Proceedings of the Sixth Applied Natural Language Processing Conference, vol. 6 (2000), pp. 224-231.
[13] Toutanova, K.; Klein, D.; Manning, C. D.; Singer, Y. Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, (2003), pp. 173-180.
[14] Gießbrecht, E.; Evert, S.: Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus. In: Proceedings of the 5th Web as Corpus Workshop, vol. 5 (2009).
[15] Robacker, F. J.: A Comparison of Five Readability Indexes. Pennsylvania State University, 1970.