Quality Indicators for Text Data

Cornelia Kiefer[1]

Abstract: Textual data sets vary in terms of quality. They have different characteristics such as the average sentence length or the amount of spelling mistakes and abbreviations. These text characteristics have influence on the quality of text mining results. They may be measured automatically by means of quality indicators. We present indicators, which we implemented based on natural language processing libraries such as Stanford CoreNLP[2] and NLTK[3]. We discuss design decisions in the implementation of exemplary indicators and provide all indicators on GitHub[4]. In the evaluation, we investigate free texts from production, news, prose, tweets and chat data and show that the suggested indicators predict the quality of two text mining modules.

Keywords: data quality, text data quality, text mining, text analysis, quality indicators for text data

1 Introduction

Plenty research on the quality of structured data tries to capture the quality of a data set as a number, e.g., in the interval [0,1] where 0 means bad quality and 1 means high quality (e.g., see [SC13], [CKK17]). E.g., the percentage of null, out-of-domain and duplicate values indicate the quality of structured data sets and can be expressed as a number in [0,1]. These methods are based on a comparison of the structured data to a 'perfect' version of the data (or parts of it) that represent the real world, or to a rule that captures characteristics of such perfect versions of the data set. Unstructured textual data needs to be processed in natural language processing pipelines. Thus, additional means to capture how text characteristics influence the quality of text analysis modules in such pipelines are needed. However, corresponding methods for texts are missing [BS16]. The indicators suggested in this work may automatically measure text characteristics such as the percentage of spelling mistakes, the number of unknown words and the confidence of standard text processing tools. Quality indicators for text data are needed: the amount of unstructured text data is exploding. Moreover, text data in science, humanities and industry comprise various crucial information such as descriptions of experimental settings, experience reports, error reports, and machine documentations [Ka14, LW16].

[1] University of Stuttgart, Graduate School of Excellence Advanced Manufacturing Engineering, Nobelstr. 12, Germany cornelia.kiefer@gsame.uni-stuttgart.de

[2] https://stanfordnlp.github.io/CoreNLP/

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

[4] https://github.com/kieferca/quality-indicators-for-text

doi:10.18420/btw2019-ws-15

These textual data sets differ significantly in quality. Moreover, many crucial textual data sets in science, humanities and industry are of low quality (e.g., see [KM16]). While the domain experts read parts of the textual data and are thus often aware of quality problems, concrete implemented indicators, which can be used to characterize the textual data sets are missing. Moreover, the influence of certain text characteristics on the quality of text mining results can only be discussed if concrete indicators are available. By now it is not clear what data indicators are useful and how they capture the quality of different textual data sets.

A worker or analysts who is reading a text full of spelling mistakes and abbreviations may have problems to understand it. Also, a text of bad quality may result in bad quality text mining results. The quality of such text mining results can only be calculated if manual annotations are available. Usually this is not the case for data sets in industry, science and humanities. Nevertheless, the quality of operational data sets may be indicated by means of the data quality indicators presented in this work. This paper has two **main contributions** that are facing these challenges: (1) we present 9 quality indicators for texts and (2) we investigate to what extend the suggested indicators are able to predict the quality of text analysis results of a language identifier and a part of speech tagger.

We start this paper with a presentation of related work in Section 2. Then, we list concrete quality indicators and discuss design decisions in the implementation (Section 3). In Section 4 we present and characterize the data sets used in the evaluation. Finally, we test the suggested quality indicators and present the results in Section 5. We conclude our work in Section 6.

2 Related Work

Many data quality indicators for structured data exist (e.g., [SC13, WS96]). Moreover, many works present first conceptual ideas for data quality methods for text [Sc12, BS16, So04]. However, none of these works presents quality indicators with concrete implementations that are applied to texts.

Data quality research on text is still in its beginnings, but the quality of textual documents is already considered in other research areas and applications. For example, the quality of written student essays [MK00] and of posts in online discussions [WGM07] can be assessed automatically. Also, many companies provide guidelines in writing texts such as error reports. The guidelines ensure that the texts can be processed automatically in high quality, e.g., by machine translation systems [Ku13]. For example, very long and nested sentences should be avoided with respect to the quality of an automatically generated translation of a text. Researchers on text simplification develop automated methods to simplify texts [Sh14]. Genova et al. suggest a framework to measure and improve the quality of textual specifications for software [Gé13]. While these works provide interesting starting points with respect to text data quality, they do not provide means to characterize the quality of textual data sets with respect to the quality of text analysis modules, such as the language

identifier and part of speech tagger as considered in this work. If indicators are suggested at all, these are special to the respective domain and no implementation details are given.

Readability measures such as the Flesch readability index capture how easy and fast a human may read and understand a text. Flesch's formula is based on the number of words per sentence and the number of syllables per word. For an overview on readability indices, see Klare [Kl74]. Many automatic readability checkers exist[5]. For these tools, no implementation details are given, though, and the code is closed-source. These readability measures only capture a very limited set of text characteristics, namely the number of words, syllables and sentences.

Particularly in the medical domain, much work is done in automatically detecting and resolving abbreviations (e.g., see [Li18]). In these works, the focus lies on resolving abbreviations and the percentage of abbreviations is not used as quality indicator. Botha et al. [BB12] investigate the effect of text size on the accuracy of language identifiers. They found that, the smaller the text size, the lower accuracy is. In our evaluation, we confirm this result. Additionally, we add more indicators besides text size and investigate the accuracies of a language identifier as well as a part of speech tagger.

While many valuable first reference points for quality indicators for text data exist, they do not cover all necessary aspects. They are oftentimes not executable or closed-source and come from fields different than data quality research and thus have a limited perspective on text quality. Moreover, in none of these related works indicators are applied to various data sets of varying quality as characterized by text analysis modules.

3 Quality Indicators for Text Data

In a text analysis pipeline the raw textual data is processed by several text analysis modules such as a language identifier, a part of speech tagger and a named entity recognizer. These modules enrich the textual data with information on the language a text is written in, the parts of speech of the words such as *verb* and *noun* and with named entities such as on companies, countries and persons. Thus, a reasonable measurement of the quality of texts needs to consider two main components: (1) the raw text data and (2) the text analysis modules. The latter group of indicators measure text characteristics with respect to standard text analysis modules, which employ default resources (such as newspaper texts as training data). These standards and defaults are oftentimes employed in domain-specific text analysis pipelines.

In Table 1, we present a non-exhaustive list of quality indicators for textual data. We restrict the methods presented to those applicable to textual data in the context of text analysis. All indicators are freely available on GitHub[6]. In the following, we describe an

[5] e.g., hemingwayapp.com and readable.io
[6] https://github.com/kieferca/quality-indicators-for-text

exemplary excerpt of the indicators and give design decisions in the implementation. The implementation of the indicator 'percentage of abbreviations' is based on a supervised machine learning algorithm and is more complex. All other quality indicators listed in Table 1 have straightforward implementations which are based on existing natural language processing libraries.

Tab. 1: Text Data Quality Indicators with respect to Data and Text Analysis Modules

Group	Indicator ID	Indicator Description
Data	1	Percentage of abbreviations
	2	Percentage of spelling mistakes
	3	Lexical diversity
	4	Percentage of uppercased words
	5	Percentage of ungrammatical sentences
	6	Average sentence length
Text Analysis Modules	7	Fit of (default) training data
	8	Confidence of standard processing modules
	9	Percentage of unknown words

The implementation of the first indicator, which automatically measures the **percentage of abbreviations (indicator 1)** is based on the Stanford Named Entity Recognizer[7]. This is a classifier which automatically recognizes named entities such as persons, cities and companies. Therefore, it uses information gained via natural language processing, such as the part of speech tags and syntax. Also, it uses training data manually annotated with named entities. It is based on conditional random fields (CRF), a supervised machine learning algorithm for sequential classifications[8]. In our case, the sequence to classify is a sequence of words. Given the sequence of words, the method classifies each word as abbreviation or non-abbreviation. To adapt the Stanford NER classifier to the task of determining if a word is an abbreviation or not, we trained it on a new training data set, which we compiled by manually annotating all individual words in a text collection with the two labels abbreviation and non-abbreviation. The compiled training data set is based on annotated excerpts of the data sets listed in Section 4. We moreover adapted the Stanford Named Entity Recognizer to the task of detecting abbreviations by implementing additional features, which are based on natural language processing methods from Stanford CoreNLP[9]: (1) word length, (2) contains symbols, (3) contains period, (4) sentence dependencies, (5) sequence of vowels and consonants representing the current word and (6) wordform (sequence of upper and lowercased characters representing the current word). We evaluated the classifier prototype on unseen data resulting in a precision of 0,85 and a recall of 0,72. Thus, it works reliable enough for our purpose of measuring the percentage of abbreviations as data quality indicator. For the calculation of precision and recall, we used the data sets as described in Section 4 and split them into separate training and testing slices. In Section 5

[7] https://nlp.stanford.edu/software/CRF-NER.shtml
[8] The CRF sequence models used are described in [FGM05].
[9] https://stanfordnlp.github.io/CoreNLP/

we will investigate if the percentage of abbreviations in a text is useful in predicting it's quality.

The **percentage of spelling mistakes (indicator 2)** in a text corpus may be calculated using the Python implementation PyEnchant[10] or any other spelling correction module.

Lexical diversity (indicator 3) is calculated using standard methods in NLTK for counting words. It is based on a formula suggested in the NLTK book [BKL09]. The relevant code is displayed in Listing 1, where the length (len) of the set of all tokens and words in the text (set) is divided by the length of all tokens and words in the text.

```
1 def lexical_diversity(text):
2 return (len(set(text)) / len(text))
```

List. 1: 'Lexical diversity' implementation based on standard Python tools

For measuring the **fit of (default) training data (indicator 7)**, we calculate the text similarity of the operational text data set that is actually being analyzed and the default training data set. Since it is most often used as default in many processing modules in natural language processing, we use the Treebank data set as default (see Section 4). We employ the Cosine Similarity metric from the DKPro Similarity library[11]. The core method used is illustrated in Listing 2. The whole concept, design decisions in implementation and a throughout evaluation with various text similarity metrics will be presented in future work.

```
1 TextSimilarityMeasure measure = new CosineSimilarity();
2 double score = measure.getSimilarity(operational, default);
```

List. 2: Excerpt of 'Fit of default training data' implementation based on DKPro Similarity

The **confidence of standard processing modules (indicator 8)** can be calculated for many classifiers, e.g., for the part of speech tagger. A statistical classifier estimates the probabilities for each class from a fixed list of classes. These probabilities are also called confidence values (for more details, see [GFL06]). Confidence is expressed as a number in the interval [0,1]. For example, confidence measures are available and can be retrieved for the natural language processing tools in OpenNLP[12] (such as the tokenizer and part of speech tagger). To get these confidence values, we followed the documentation of the OpenNLP library (see footnote 12). E.g., for the part of speech tagger, we just call the *probs* method which returns an array of the probabilities for all tagging decisions. The method is shown in Listing 3. Then, we calculate the mean over all sentences and return it. In Section 5 we discuss if these confidence values for the OpenNLP part of speech tagger may be used as a quality indicator.

[10] http://pythonhosted.org/pyenchant/
[11] https://dkpro.github.io/dkpro-similarity/
[12] https://opennlp.apache.org/

```
1 POSTaggerME tagger = new POSTaggerME(model);
2 tagger.tag(sentence);
3 double probs[] = tagger.probs();
```

List. 3: Excerpt of 'confidence of standard processing modules' implementation based on OpenNLP

The **percentage of unknown words (indicator 9)** may be calculated by applying the standard part of speech tagger implemented in NLTK to the texts, which has an individual class for unknown words, i.e., 'X'.

The measured percentages and raw numbers need to be transferred into consistent data quality metrics in [0,1] where 0 means low and 1 high quality. We transfer the measured percentages and raw numbers by means of adequate step functions. Some indicators such as 'confidence' are already fitting numbers in [0,1] and do not need to be transferred. But other indicators such as the percentage of abbreviations and the average sentence length need to be transferred into a consistent quality metric in [0,1]. For example, the first indicator measures the percentage of abbreviations. A high percentage of abbreviations should result in a low quality metric and a low percentage of abbreviations in a high quality metric. This can be achieved by means of a step function, where, e.g. 0-1% abbreviations are transferred to the quality metric 1 and >10% to 0, etc. We will describe this transfer of indicators to data quality metrics in more detail in future work.

The indicators presented build the basis for methods that can improve the quality of texts. These will be adressed in future work. For example, if the measured percentage of spelling mistakes is high, data quality may be improved by means of an automatic spelling mistakes correction method.

4 Data Sets used in the Evaluation

We conduct experiments on 5 different data sets. They comprise prose, news, chat posts, tweets and production data. The prose and news data sets (Brown and a subset of the Penn Treebank) and chat posts (NPS Chat data) are taken from NLTK[13]. The Twitter corpus was taken from Gimpel et al. [Gi11]. Additionally, we employ a confidential data set from an industry partner in Germany. It comprises information on downtimes in a production line and contains German free text information. The data set contains information on the reasons for downtimes and the actions that were taken to put the production line running again. The workers on the shop floor can fill the free text field via text entry into a tablet. In Table 2 we list the main characteristics of the data sets. All data sets come with gold annotations for at least one text mining module. Thus, in our evaluational setting, we are able to calculate accuracies. Accuracy calculations will be discussed in the next section.

[13] http://www.nltk.org/nltk_data/

Tab. 2: Data sets used in the experiments

Data collection	Type	# of tokens
Brown	Prose	1.15M
Treebank (stub)	News	40k
Twitter corpus	Tweets	35k
NPS Chat	Chat	45k
Industry corpus	Production	153k

5 Evaluation

The quality of text mining results is judged by comparing the predictions of the tools with the gold labels annotated by human experts. For example, to determine the quality of a part of speech tagger, it's 'Token Accuracy' is calculated as shown in Equation 1.

$$ACC = \frac{(\#\ correct\ POS\ tags\ in\ tagged\ data)}{(\#\ total\ POS\ tags\ in\ tagged\ data)} \tag{1}$$

The accuracy of language identifiers is calculated by comparing the gold language annotations with the annotations made by the tool. As already mentioned in the introduction of this work, accuracies can only be calculated if manual annotations are available. This is oftentimes not the case. The calculation of the suggested quality indicators does not need such manual annotations, though. In Table 3, we show first results with respect to whether they are able to predict the quality of such tools.

In the first column in Table 3, we note the data set. In the following three columns we present the overall and single accuracies for two text analysis modules: (1) the Apache Tika language identifier[14] (LI (Tika)) and (2) the CRF part of speech tagger from NLTK[15] (POS (CRF)). Compiling manual annotations costs time and expert knowledge. Therefore, as oftentimes the case for operational data sets, for the industry data no manual annotations of part of speech are available. Thus, part of speech (POS) accuracy can't be calculated. Nevertheless, the accuracy of the language identifier (LI) and the indicators give insights on the textual characteristics. In the following columns, we present the results for our suggested quality indicators for text data.

We report the raw numbers gained for data quality indicators as described in Section 3. Thus, most indicators are measured in percent and some are plain numbers such as the average sentence length. In future work, these raw measurement results need to be transferred to uniform data quality metrics as already mentioned in Section 3. Also, further analysis modules, implementations and data sets need to be adressed in future work.

From first to last row, the overall accuracy of the two text mining modules decreases. Treebank and Brown (news and prose) can be processed in a reliable quality by these text

[14] https://tika.apache.org/

[15] https://www.nltk.org/_modules/nltk/tag/crf.html with the universal tagset and Treebank training data

Tab. 3: Evaluation results

Data	Accuracy			Indicator								
	Overall	LI (Tika)	POS (CRF)	Abbreviations (1)	Spelling (2)	Lexical Diversity (3)	Uppercased (4)	Ungrammatical (5)	Avg. Sentence Length (6)	Fit of training data (7)	Confidence (8)	Unknown words (9)
Treebank	0,90	0,86	0,94	2,0	19,0	1,5	1,6	0,1	24,0	1,0	0,9	0,0
Brown	0,86	0,84	0,88	0,6	12,0	0,1	0,9	0,4	20,3	0,9	0,9	0,1
Twitter	0,62	0,47	0,76	4,6	27,0	10,6	7,0	1,5	14,5	0,5	0,8	0,2
Chat	0,49	0,20	0,78	11,0	34,0	4,6	15,8	1,0	4,3	0,5	0,6	0,3
Industry	n.a.	0,34	n.a.	7,1	23,0	0,4	0,1	0,0	4,8	0,5	0,7	0,5

mining tools (=Overall Accuracy is high). Tweets, chat posts and industry data can only be processed in low quality (=Overall Accuracy is low). A similar classification is made by the data quality indicators: Treebank and Brown contain less abbreviations and spelling mistakes and have a low lexical diversity. The amount of uppercased characters is low. They hardly contain ungrammatical sentences. The sentences are longer when compared to tweets and especially when compared to chat and industry data. The fit of training data and confidence are high, and the amount of unknown words is low.

Low quality of Tweets, Chat posts and Industry data is indicated by many abbreviations, spelling mistakes and unknown words as well as a low fit of training data and low confidence values. Tweets contain a significantly higher lexical diversity than the other data sets. Chat posts contain particularly many abbreviations and lexical diversity is high. In both, Tweets and Chat posts, more words than in the other data sets are uppercased and they contain more ungrammatical sentences. In Chat and Industry data the sentences are very short. The industry data is full of domain-specific abbreviations, unknown words and spelling mistakes. Lexical diversity is rather low and the sentences are very short and parseable, i.e. the number of ungrammatical sentences is low.

Both text analysis modules selected are high quality modules oftentimes employed in text mining projects in industry. While the language identifier seems to be very sensible with respect to some text characteristics, the part of speech tagger is more robust. From Table 3 it can be seen that the suggested indicators are good starting points that may indicate quality. Thus, in a real analysis situation in humanities, science or industry, where accuracies are not calculable and thus not known, the suggested data quality indicators give a hint on how good processing modules may be able to cope with the data set(s).

6 Conclusion

We have presented 9 data quality indicators for text data sets. Operational text data sets usually do not come with manual gold annotations for text processing steps. Thus, the quality of many text analysis results is not known in text mining projects in the humanities, science and industry. We suggested data quality indicators which help in deciding if default text mining modules will deal easily with the textual data or not, i.e. if improvement strategies are needed or not. For each indicator, corresponding improvement strategies exist, which will be adressed in future work. Moreover, in future work we adress the transformation of percentages and raw numbers into data quality metrics in [0,1] and integrate and combine the methods into a complete framework for data quality assessment and improvement.

Acknowledgment

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

References

[BB12] Botha, Gerrit Reinier; Barnard, Etienne: Factors that affect the accuracy of text-based language identification. Computer Speech & Language, 26(5):307–320, 2012.

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

[BS16] Batini, Carlo; Scannapieco, Monica: Data and Information Quality. Springer International Publishing, Cham, 2016.

[CKK17] Chung, Yeounoh; Krishnan, Sanjay; Kraska, Tim: A Data Quality Metric (DQM): How to Estimate the Number of Undetected Errors in Data Sets. Proc. VLDB Endow., 10(10):1094–1105, 2017.

[FGM05] Finkel, Jenny Rose; Grenager, Trond; Manning, Christopher: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 363–370, 2005.

[Gé13] Génova, Gonzalo; Fuentes, José Miguel; Morillo, Juan Llorens; Hurtado, Omar; Moreno, Valentin: A framework to measure and improve the quality of textual requirements. Requir. Eng., 18(1):25–41, 2013.

[GFL06] Gandrabur, Simona; Foster, George; Lapalme, Guy: Confidence Estimation for NLP Applications. ACM Transactions on Speech and Language Processing (TSLP), 3(3):1–29, 2006.

[Gi11] Gimpel, Kevin; Schneider, Nathan; O'Connor, Brendan; Das, Dipanjan; Mills, Daniel; Eisenstein, Jacob; Heilman, Michael; Yogatama, Dani; Flanigan, Jeffrey; Smith, Noah A.: Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 42–47, 2011.

[Ka14] Kassner, Laura; Gröger, Christoph; Mitschang, Bernhard; Westkämper, Engelbert: Product Life Cycle Analytics - Next Generation Data Analytics on Structured and Unstructured Data. In: Proceedings of the 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14. Elsevier, Naples, pp. 1–6, 2014.

[Kl74] Klare, George R.: Assessing Readability. Reading Research Quarterly, 10(1):62–102, 1974.

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

[Ku13] Kuhn, Tobias: A Survey and Classification of Controlled Natural Languages. Computational Linguistics, 40(1):121–170, 2013.

[Li18] Liu, Yue; Ge, Tao; Mathews, Kusum S.; Ji, Heng; McGuinness, Deborah L.: Exploiting Task-Oriented Resources to Learn Word Embeddings for Clinical Abbreviation Expansion. CoRR, abs/1804.04225, 2018.

[LW16] Lemke, Matthias; Wiedemann, Gregor: Text Mining in den Sozialwissenschaften. Springer Fachmedien, Wiesbaden, 2016.

[MK00] Miltsakaki, Eleni; Kukichy, Karen: Automated evaluation of coherence in student essays. In: Proceedings of LREC, pp. 1–8. 2000.

[Sc12] Schmidt, Andreas; Ireland, Chris; Gonzales, Eloy; Del Pilar Angeles, Maria; Burdescu, Dumitru Dan: , On the Quality of Non-structured Data, 2012.

[SC13] Sebastian-Coleman, Laura: Measuring data quality for ongoing improvement: A data quality assessment framework. Elsevier Science, Burlington, 2013.

[Sh14] Shardlow, Matthew: A Survey of Automated Text Simplification. International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing 2014, 4(1), 2014.

[So04] Sonntag, Daniel: Assessing the Quality of Natural Language Text Data. In: GI Jahrestagung. pp. 259–263, 2004.

[WGM07] Weimer, Markus; Gurevych, Iryna; Mühlhäuser, Max: Automatically Assessing the Post Quality in Online Discussions on Software. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. ACL '07, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 125–128, 2007.

[WS96] Wang, Richard Y.; Strong, Diane M.: Beyond accuracy: what data quality means to data consumers. J. Manage. Inf. Syst., pp. 5–33, 1996.