Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Automatic Linguistic Analysis of Annotated Text to Improve Standards Development

Mario Fusani[a], Giuseppe Lami[b, *] and Francesco Merola[b]

[a] *Marconi Labs Coltano, Pisa, Italy*

[b] *Istituto di Scienza e Tecnologie dell'Informazione Consiglio Nazionale delle Ricerche Pisa, Italy*

ARTICLE INFO

Article History:
Submitted 4.8.2025
Revised 8.8.2025
Second Revision 8.20.2025
Accepted 9.8.2025

Keywords:
Natural Language Processing
Standards Development
Annotated Text

ABSTRACT

The draft Standards in progress, when accompanied by reviewers' comments, represent a clear example of annotated textual data and are therefore attractive for "supervised machine learning" analyses. This article describes an ongoing experiment in which a collection of annotated drafts is analyzed using automatic linguistic analysis techniques to identify effective solutions for improving the Standards creation process. The goal is to predict the parts of the standard that are prone to comments. The results achieved so far are presented and discussed along with new possible directions to take.

1. Introduction

The recognized international Standards result from intense activity regulated by defined and approved processes. These processes generate technical documents created by expert working groups in various fields and endorsed by recognized organizations such as ISO, IEC, IEEE, and others [1].

Critical phases in the Standard creation process are the recurrent expert reviews that the drafts of Standard documents undergo before final approval. Expert reviewers of draft standards are required to propose changes, classify them according to the issue they contain (typically, an issue can be classified as General, Technical High, Technical Low, and Editorial), and provide comments describing the found issue and, possibly, proposing a better wording. The standard draft review is a time-consuming activity with a significant impact on the time-to-release of standards.

Providing standardization bodies with effective techniques and tools to support the expert review process would provide benefits for standards editors and users.

*Corresponding author
Email address: giuseppe.lami@isti.cnr.it
Website: https://www.isti.cnr.it/it/chi-siamo/people-detail/186/Giuseppe_Lami
ORCID: 0000-0003-2960-5241

This paper reports the results of a research activity aimed at exploring the application of linguistic techniques to analyze "draft-and-comments" document pairs to early identify critical sentences (i.e., sentences that will be likely commented by expert reviewers). Early identification of critical sentences may support human reviewers, speed up the review process, and enhance the quality of final standards. The purpose is to improve the review process by determining possible characteristics of the sentences that could, in some way, justify the presence of comments from expert reviewers. The exploratory research involved the application of two different approaches of Natural Language Processing [2]. These two approaches are described in this paper along with experimental results on a set of real standard drafts and related comments. The first approach consists of automated checking and reporting of possible quality issues of clauses in a draft Standard text. The objectives of such an approach are the identification of possible linguistic flaws in the standard clauses, and the verification of the degree to which the comments of expert reviewers are related to such quality issues. In the rest of the paper, his approach is identified as Linguistic Check.

The second approach presented in this paper aims at determining, through Machine Learning (ML) techniques [3], the likelihood that each requirement or clause of a draft Standard will receive comments from the expert reviewers. The objective of such an approach

DOI reference number: 10.18293/JVLC2025-N2-076

is to identify possible general flaws in the clauses. This approach is identified as Data-based check.

To investigate the feasibility of the Linguistic and Data-based Checks and gather some evidence on their effectiveness, in this study they have been applied to the same set of commented draft standards.

The draft standards and the related comments are confidential. To preserve their confidentiality, the contents of these documents are not disclosed in this paper.

This paper is structured as follows: in Section 2, the preparation activities required for the application of the presented approach are described. In Section 3, the Linguistic Check approach is described, and the results of its application are presented and discussed. In Section 4, the Data-based Check approach is described, and the results of its application are presented and discussed. Finally, in Section 5, conclusions and the description of the next steps are provided.

2. Context of the Study and Data Preparation

The material used in the study consists of 6 safety-related draft Standards, along with related expert comments. Although this raw data may be too limited to derive high-confidence statistics, it is used to set and test all preprocessing and processing apparatuses (see Sections III and IV).

The characteristics of the data used in the study make them particularly suitable for the analysis performed. They are homogeneous in terms of content as they are technical standards addressing the same discipline (i.e., functional safety) with a generally accepted reference vocabulary. The level of expertise and knowledge of the reviewers is generally high, so the technical level of the comments is expected to be homogeneous as well.

The quality of safety-related standards has long been a subject of attention [10, 13], due to their great impact on people's lives. As already mentioned, this work addresses both the problem of language quality, as in previous studies [11, 12, 14], this time using the experts' annotations as a comparison (Linguistic approach), and the problem of quality in general (Data-based approach).

We acknowledge that the dataset is rather small. This is because the process of obtaining the source data is quite long and complex. Working documents issued in the development of a standard (drafts and comments) are confidential and must not be accessible to people outside working groups or authorized expert reviewers. Necessary (but not always sufficient) conditions for obtaining them are:

- Be part of at least one expert working group or the authorized set of reviewers.
- Obtain authorization from the convenors of the working groups.
- Declare that the data will be used solely for research purposes to improve the quality of the standardization process.
- In any publications, do not include any part of the draft texts or comments, not even as examples, and treat data-related information anonymously.

2.1 From Raw Data to Structured Dataset

Data is automatically extracted from our source documents (consisting of PDF files in the case of draft Standards, and text cells of MS Excel files in the case of expert comments) as:

- List of clauses in draft Standards (PDF Documents), each tagged with its number.
- Lists of sentences taken from Requirements (contained in the clauses), each tagged with its line number.
- Lists of comment texts, each tagged with both its line number and corresponding clause.

The line number acts as a link between draft Standard sentences and their related comments. This allows each clause or sentence to be associated with zero or more comments.

These sets of data can be considered an annotated data set, where the data is represented by the text corresponding to the various sentences/clauses (or, as we will see, their properties), and the comments are annotations or "labels". The available data sets are then suitable for ML analytic methods (as supervised ML) applied in the Data-based check.

We should determine which features do characterize the text that could, in some way, justify the presence of comments, to gain the capability to predict the occurrence of comments if new, not-yet-reviewed, drats are made available. Extracting text at both the clause and sentence level provides flexibility and allows deeper analysis (specifically during the Data-based check approach), as will be shown in the experiment description.

The two approaches mentioned in Section I differ from each other as the data features are obtained differently. In the Linguistic check approach, the data features are a set of quality issues detected for each sentence by a tool called QuARS [4]. In the Data-based check approach, the data features are computed as the frequency distribution of a set of selected words, taken from all the words that appear in the entire collection of drafts, within each individual sentence/clause.

3. Linguistic Check: Procedure and Outcomes

In the literature, several studies addressing the Linguistic Check of requirements or standards exist; in

	sent_lines	VAGN	SUBJ	MULT	OPTN	WEAK	IMPL	USPC	DSUM	COMM	EDIT	SLEN
0	Standard-1 0 427	0.023810	0.000000	0.047619	0.000000	0.023810	0.023810	0.000000	0.119048	0.000000	0.000000	42.0
1	Standard-1 1 430	0.019231	0.000000	0.038462	0.000000	0.019231	0.019231	0.000000	0.096154	0.000000	0.000000	52.0
2	Standard-1 2 434	0.000000	0.000000	0.095238	0.000000	0.000000	0.047619	0.000000	0.142857	0.047619	0.000000	21.0
3	Standard-1 3 435	0.045455	0.000000	0.090909	0.000000	0.000000	0.000000	0.000000	0.136364	0.000000	0.000000	22.0
4	Standard-1 4 437	0.000000	0.000000	0.068966	0.000000	0.034483	0.000000	0.034483	0.137931	0.034483	0.000000	29.0
...	...	...	...	...	...	...	...	...	...	...	...	...
3151	Standard-6 580 1514	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.030303	0.030303	0.030303	0.030303	33.0
3152	Standard-6 581 1517	0.000000	0.000000	0.040000	0.000000	0.000000	0.000000	0.000000	0.040000	0.020000	0.020000	50.0
3153	Standard-6 582 1521	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	20.0

```
# Legend:
# sent_lines = Sentence ID: Standard name (anonymized) + entry num + line num in pdf document
# VAGN, SUBJ, ... USPC = QuARS defect densities per sentence
# DSUM = Feature: defect density sum per sentence
# SLEN = Feature: sentence length (number of words)
# COMM = comment density per sentence (number of comments/sentence length)
# EDIT = only editorial-type comment density per sentence
```

Figure 1: Data frame for quality analysis: defect and comment densities in sentences.

[5], a survey of these studies is provided. [6] describes the results of a study on requirements analysis that follows an approach similar to the one described in this paper that uses the QuARS tool.

The QuARS tool, developed by CNR, has been used for many years to highlight quality issues, such as ambiguities, in requirements. It can perform both lexical and syntactic analyses of sentences, detecting potential linguistic defects based on a syntax parser. Table 1 presents the range of quality issues that QuARS is capable of identifying.

Table 1: Kinds of Non-Quality characteristics detectable by QuARS

Characteristics	*Examples*
Optionality (OPTN)	"this", "if needed", …
Subjectivity (SUBJ)	"simple", "known", ...
Vagueness (VAGN)	"adequate", "easy", ...
Weakness (WEAK)	"can", …
Implicitly (IMPL)	"the previous task", "it", ...
Under-specification (USPC)	"The manual is ...",
Multiplicity (MULT)	"< expression> and / or <expression>", ..

In our experiment, QuARS was fed with 3,154 sentences from the available draft Standards. Adopting "sentence defect density" metrics (number of defects in sentence/number of sentence words), seven values have been computed using QuARS (and by some pre- and post-processing) for each sentence in the data set. Similarly, a "comment density" value was obtained for each sentence. This allowed us to get a dataset in the format of Python Pandas' "data frame" [7]. Fig. 1 shows an excerpt of it. To note that the initial experimental set of sentences is turned into a database composed of numerical items corresponding to measures associated with each sentence.

3.1 Using QuARS Results

Considering our data frame, we have 3.154 entries (sentences), of which 27% are commented.

Table 2: Ratios of defective and commented sentences

Characteristic	*Ratio*
VAGN	0.29
SUBJ	0.03
MULT	0.72
OPTN	0.02
WEAK	0.14
IMPL	0.22
USPC	0.13
DSUM	0.86
COMM	0.27

All the percentages of QuARS-reported defects are also reported in Table 2 (see the Legend in Table 1 for the abbreviations). Table 1 contains also the values of two metrics calculated from the analysis of the sentences: DSUM representing the defect density sum per sentence, and COMM representing the average comment density per sentence.

We see that Multiplicity (MULT) has quite a high rate: this is due to an excessive percentage of false positives produced by QuARS for this characteristic.

3.2 Searching for Correlations

To determine the extent to which the experts' comments reflect QuARS defects, a correlation is

calculated between the measures of QuARS defects and comment densities. Fig. 2 shows the correlation matrix (Spearman correlation was applied as the distribution of defective sentences and commented sentences may be considered to be monotonic). Just a glance at the defect density sum column (DSUM) across the COMM and EDIT rows indicates that there is no significant correlation between reported defects and comments.

	VAGN	SUBJ	MULT	OPTN	WEAK	IMPL	USPC	DSUM	COMM	EDIT
VAGN	1.000000	-0.010932	-0.040033	-0.002726	0.006571	-0.029132	0.008795	0.310645	0.001297	0.015298
SUBJ	-0.010932	1.000000	0.019587	0.019833	0.051358	-0.010364	0.023054	0.133325	0.002149	0.016761
MULT	-0.040033	0.019587	1.000000	-0.013900	0.017680	0.073436	0.013365	0.742966	-0.046564	-0.031833
OPTN	-0.002726	0.019833	-0.013900	1.000000	0.060244	-0.005376	0.044213	0.052529	-0.005953	0.001060
WEAK	0.006571	0.051358	0.017680	0.060244	1.000000	0.025721	-0.012973	0.247762	0.006697	-0.005325
IMPL	-0.029132	-0.010364	0.073436	-0.005376	0.025721	1.000000	0.022964	0.360814	-0.016386	-0.035755
USPC	0.008795	0.023054	0.013365	0.044213	-0.012973	0.022964	1.000000	0.238168	-0.008651	0.000341
DSUM	0.310645	0.133325	0.742966	0.052529	0.247762	0.360814	0.238168	1.000000	-0.057038	-0.050415
COMM	0.001297	0.002149	-0.046564	-0.005953	0.006697	-0.016386	-0.008651	-0.057038	1.000000	0.749568
EDIT	0.015298	0.016761	-0.031833	0.001060	-0.005325	-0.035755	0.000341	-0.050415	0.749568	1.000000

Figure 2: Correlation matrix among QuARS defect and comment densities.

3.3 Outcomes of Linguistic Check Approach

The comments from expert reviewers do not align with the linguistic defects reported by QuARS (as indicated in Table 1). That indicates that using tools like QuARS can effectively complement the reviewers' work.

4. Data-Based Check: Procedure and Outcomes

Another way to find features that allow us to predict whether a sentence or clause will receive comments is to utilize the same words it contains, without considering their meaning.

More precisely, we adopt a technique similar to the one called “bag of words” (BOW) [8]: That is, we select the N most frequent, non-banal words from those of the whole draft standard collection; then, for each sentence or clause, we compute how many times each of these words appears. This results in an N-dimensional vector representation, which serves as an alternative feature set compared to the one used in the previous approach.

This process produces two data frames: one organized by sentences and the other by clauses, shown in Table 3 and Table 4: 4, respectively. Starting from Table 3, the sentence ID in the first column has two terms: a progressive index and the index of the original set. They differ because it was decided to remove, from the original set, the sentences longer than L words (L being a parameter varying between 60 and 120). In our experiment, this data set therefore has 3074 different sentences (out of the total 3154 sentences: the former index appears as a second term in the ID). Columns from the second to the fourth, respectively, contain the number of total comments, the number of editorial comments, and a code (1 to 6) anonymously associated with each Standard. This code is used by the function that "stratifies" the dataset when it gets partitioned, so that the Standards are proportionally represented across the partitions. The features are the N-element vectors mentioned above, expressed as floating-point numbers just for computational reasons.

Table 4 follows the same structure, with the sole difference that the first column contains a single ID, as all clauses were maintained.

Table 3: Data frame with frequent word counts as features - Sentences

SENT_ID	COMM	EDIT	STD_No.	FEATURES
0 0	0	0	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
1 1	1	1	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, …
2 2	1	0	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, …
3 3	0	0	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
…	..	..	..	…
3071 3151	1	1	6	[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, …
3072 3152	1	1	6	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
3073 3153	0	0	6	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …

Table 4: Data frame with frequent word counts as features - Clauses

CLAUS_ID	COMM	EDIT	STD_No.	FEATURES
0	0	0	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
1	0	0	1	[3.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, …
2	0	0	1	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, …
3	0	0	1	[2.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, …
…	..	..	..	…
985	3	0	6	[0.0, 2.0, 0.0, 0.0, 0.0, 1.0, 1.0, 2.0, 1.0, …
986	0	0	6	[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, …
987	0	0	6	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …

Moreover, it was decided to simplify the experiments by considering the labels as binary values instead of the number or density of comments. The new labels indicate whether any sentence/clause is commented on or not, without distinguishing between editorial or technical comments.

For all the experiments discussed in the following sections, the data set has been partitioned into a training set and a test set (the test set ratio, set to 30%, being an experiment parameter) after shuffling the rows of the data frame and performing a stratification with respect to the different standards. A portion of the entries of the training set (related to the data structured by sentences), whose features and labels parts are named, respectively, X_train and y_train, is shown in Figure 3 (note the shuffled row indices shown in the y_train part).

```
X_train[240:255]
array([[1., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 2., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

```
y_train[240:255]
1963     True
2144    False
652     False
        ...
1400    False
311     False
29      False
```

Figure 3: Portion of a training set from the data frame. "True" means "commented sentence".

The analysis focuses on the training set to evaluate the capability of predicting, based on the current features, the class (commented / non-commented) of the corresponding labels. The chosen model is a classification model called the SGD (Stochastic Gradient Descent) Classifier [9].

A cross-validation training-and-checking process is executed, in which the training set is divided into *p* partitions (where *p* is a parameter that takes the values of 3 or 4). In this process, *p* computations are done. In each computation, a different partition is put aside for validation while the others are trained. The predicted labels of the validation subset are compared with their actual labels each time, and internal scores are evaluated to select the best training. At the end of the cross-validation process, four values are generated for the whole training set:

- True Positive (TP): number of both True predicted and actual labels.
- True Negative (TN): number of both False predicted and actual labels.
- False positive (FP): number of labels predicted as True when the actual labels are False.
- False Negative (FN): Number of labels predicted as False when the actual labels are True.

Two important indicators are computed from the above values, which inform us about the effectiveness of the chosen features and model:

1) Precision, computed as TP/(TP+FP), tells how our approach is good at predicting commented sentences.
2) Recall, computed as TP/(TP+FN), tells how our approach is good at not missing commented sentences.

In the next Section, we show, by experiments, how these indicators are used to adjust the features to improve the results of the analysis.

4.1 Deriving New Features by Refining the BOW Set

The classification algorithm allows for assigning a score to each sentence/clause, so that they can be ordered from those least likely to be commented on to those most likely to be commented on.

Furthermore, the library functions associated with the model [15] provide a method to get the counts of the values of TP, FP, TN, and FN if all the labels are considered positive above a certain threshold in the score. Increasing the threshold results in a generally increased Precision and an always decreased Recall.

This approach is characterized by the iterative refinement of the N-element BOW. As previously mentioned, the initial BOW is constructed using the entire draft standard collection. In each subsequent step, the BOW is refined by considering a reduced set of sentences, specifically those that are both TP and have scores above a certain threshold.

4.2 Experiments at the Sentence Level

The classification algorithm allows for assigni
The first round of experiments was conducted at
sentence level (i.e., starting from the data frame in
3). The following parameters were chosen after
preliminary trials:

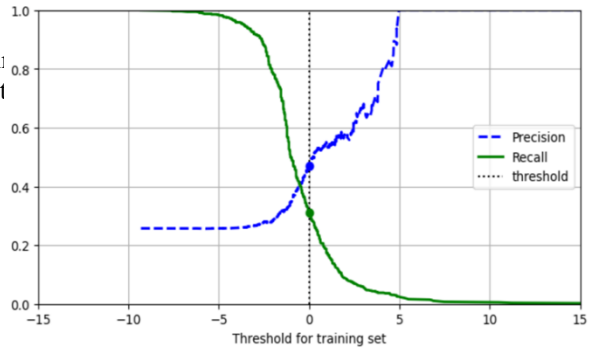
- Set of stop-words (i.e., conjunctions, prepositions, and other words without semantic meaning).
- Indices for extracting the BOW from the list of the most frequent words in a defined set S of sentences. At the beginning, S is the whole set of sentences in the draft Standard collection. At any successive step, S is changed as mentioned above.
- Test set ratio = 0.3.
- Maximum sentence length, L= 80 (consequently, 3074 out of 3154 sentences were considered).
- Number of partitions in the training set to perform cross-validation: p=3.
- Threshold values: 0, 3, 4.

The above parameters were chosen following a series of preliminary experiments. After that, they were kept fixed while the BOW was iteratively refined. During each iteration, the algorithm was run multiple times with different thresholds, and the Precision and Recall values were calculated for each run (see Fig. 4). Figures 4a) and 4b) show the variation of Precision and Recall with respect to the threshold in the iterations 0 to 2. We observe that the generally acknowledged "best results," which occur when Precision equals Recall, yet modest ones, slightly increase across the iterations.

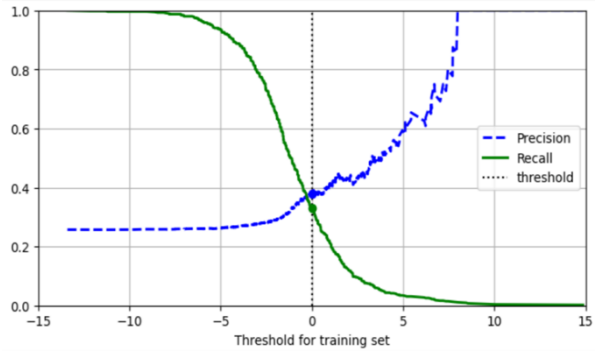
At each iteration, after the cross-validation within the training set, the trained model is run on the test set.

Table 5 shows the results of each iteration, with 0, 3, and 4 as threshold values. More precisely, it shows, in the upper part, the results of each iteration of the training phase, and in the lower part, the results of each iteration of the test phase. Based on the values of Precision and Recall appearing in Table 5, it is evident that this experiment may not produce significant results at this stage, primarily due to the limited dataset size of only 3,074 sentences. Nevertheless, the method appears promising as it demonstrates some progress in the results (Precision increases in iteration 2, yet at Recall's expense).

To continue with a new iteration, a new BOW could be extracted, for example, from the 32 sentences identified as TP scored higher than Threshold 3 in Iteration 2. Then the model would be trained using the BOW counts as features.



a)



b)

Figure 4: Precision-Recall curves obtained from experiments with the sentence-level structured data-frame: a) iteration 0; b) iteration 2.

Table 5: Results from recurrent runs – Sentence level experiments

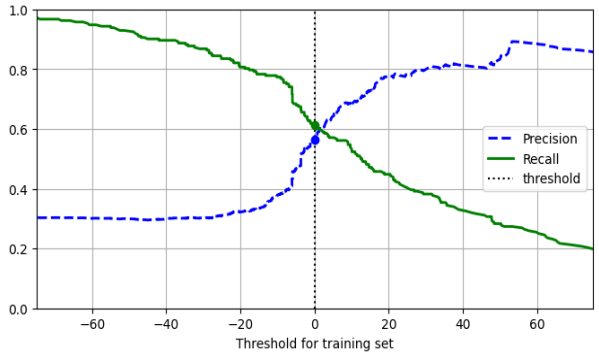
	ITERATION	0			1			2		
	Threshold	0	3	4	0	3	4	0	3	4
Training Set Validation Predictions	Precision	0.38	0.45	0.51	0.42	0.64	0.72	0.47	0.68	0.80
	Recall	0.33	0.07	0.04	0.29	0.05	0.03	0.31	0.06	0.04
	True Positive (well guessed)	182	41	24	160	29	16	172	32	21
	False Positive (wrong guessed)	298	51	23	219	16	6	193	15	5
	False Negative (missed)	371	512	529	393	524	537	381	521	532
Training Set Sentences	Total Sentences	2151								
	Commented	553								
Test Set Predictions	Precision	0.49			0.44			0.56		
	Recall	0.30			0.34			0.24		
	True Positive (well guessed)	79			89			64		
	False Positive (wrong guessed)	82			113			50		
	False Negative (missed)	182			172			197		
	Test Comment Prediction Correlation	0.21			0.19			0.23		
Test Set Sentences	Total Sentences	923								
	Commented	261								

4.3 Experiments at the Clause Level

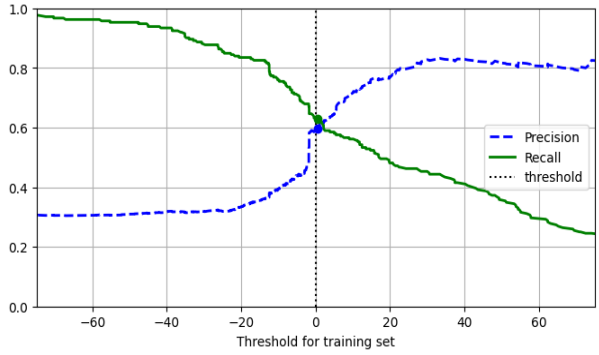
A similar set of experiments was conducted at the clause level (i.e., using the data frame in Table 4). The parameters described in Sub-section 4.2 were maintained, with the exception that the maximum sentence length was removed, as it would not make sense in the case of clauses.

Again, multiple iterations were conducted and the

BOW progressively refined. The resulting Precision-Recall curves are shown in Fig. 5 and the detailed results of all the runs are reported in Table 6 In this case, the outcomes are more encouraging: precision and recall consistently remain around 60% or slightly higher, without either metric dropping significantly as the threshold varies. Moreover, we still notice a (minor) general improvement across iterations, due to the BOW refinement.



a)



b)

Precision-Recall curves obtained from experiments with the clause-level structured data-frame: a) iteration 0; b) iteration 2.

Table 6: Results from recurrent runs – Clause level experiments

	ITERATION	0			1			2		
	Threshold	0	3	4	0	3	4	0	3	4
Training Set Validation Predictions	Precision	0.57	0.62	0.63	0.61	0.64	0.65	0.60	0.62	0.64
	Recall	0.61	0.58	0.57	0.62	0.60	0.60	0.62	0.59	0.58
	True Positive (well guessed)	130	124	121	131	127	127	133	125	124
	False Positive (wrong guessed)	100	76	71	84	716	69	90	74	70
	False Negative (missed)	82	88	91	81	85	85	79	87	88
Training Set Clauses	Total Sentences	691								
	Commented	212								
Test Set Predictions	Precision	0.67			0.61			0.62		
	Recall	0.59			0.52			0.59		
	True Positive (well guessed)	58			51			57		
	False Positive (wrong guessed)	28			32			35		
	False Negative (missed)	40			47			41		
	Test Comment Prediction Correlation	0.38			0.38			0.41		
Test Set Clauses	Total Clauses	297								
	Commented	98								

4.4 Discussion on the Outcomes of the Data-Based Check

While the amount of data currently available to us is insufficient to claim generalization of the approach, the outcomes show some positive insights.

The first consideration is about the iterative BOW refinement process, which appears to have a positive impact on the model's predictive capabilities (especially noticeable in the experiments conducted at the sentence level). This suggests that further experimentation may be worthwhile, potentially with longer iterations once additional data becomes available.

The second point concerns the difference in results when structuring the data as single sentences versus entire clauses, with the latter currently yielding better performance. This could be due to several factors. First, although reviewers' comments are associated with specific line numbers (and thus sentences), they often address a broader context that may not be fully captured within the sentence itself. A clause is more likely to provide the necessary context.

Secondly, mapping BOW-derived features to individual sentences produces very sparse vectors, as sentences are typically short. Clauses, being longer, yield denser feature vectors, which may help the model form better associations.

Finally, structuring the data by clauses results in a generally more balanced dataset, which again may benefit the model training.

Nevertheless, using sentences offers the important advantage of finer-grained output (knowing that a sentence is likely to receive a comment is more informative than knowing the same for a clause), making this approach worth further experimentation.

Overall, despite the current limitations, we could still consider utilizing the trained model in the "production stage", guiding the reviewers of a new draft standard to focus their work on true positive (TP) predicted sentences. For example, consider Table 6. If the test set were presented to a reviewer as a new standard and the 3-times-trained model were applied to it for a prediction, then a reviewer would have 297 clauses with a total of 57+35 = 92 clauses declared positive for comment. By a manual analysis of these, the reviewer would have a 62 percent chance of guessing the ones to comment on, while they would not notice 41 clauses that should be commented on but would have rightly omitted to work on 297-57-35-41 = 164 clauses.

5. Conclusions

This paper describes an exploratory study aimed at identifying techniques, based on automatic linguistic analysis, able to provide support to the technical review phase of standard development. The study focuses on the application of two classes of approaches to NLP, the Linguistic check (based on the syntax analysis of

sentences to identify linguistic quality issues) and the Data-based check (based on ML). Both approaches are applied to evaluate their capability of establishing the extent to which a sentence or clause in a draft standard is prone to being commented on by reviewers. To this aim, we used a set of draft standards, each of them associated with the comments from reviewers. The draft standards are then annotated texts, as each comment is related to a uniquely identified sentence of the draft standard.

The outcomes of the study show that in the case of the Linguistic check (applied using the QuARS tool), no significant correlations have been found between sentences containing quality issues and commented ones. That indicates that using tools like QuARS can effectively complement the reviewers' work as they address issues not detected by reviewers. In the case of Data-based check, we apply a technique that uses ML algorithms, based on BOW, iteratively refined. The application of this approach is promising, as the experimental outcomes indicate that the refinement of the BOW can produce features that enhance the model's predictive ability.

Given the exploratory nature of the study, the validity of the experimental results is still limited. Especially in the case of the Data-driven approach, the experiments can be improved in several aspects:

- consider a larger range of natural language processing techniques, algorithms, and models (as powerful classifiers or Artificial Neural Networks) to improve the exploration and get more significant outcomes.
- use of features different from single words (e.g., bi-grams, and tri-grams) to generate BOW.
- Try multiple classification instead of binary, considering as labels the number of comments per sentence and/or lexical sentiment analysis of the comments.

The previous improvements may allow the derivation of concrete guidance for standards editors and reviewers. A critical element to be addressed in the future steps to increase the validity of the results is the use of a larger set of commented draft Standards.

References

[1] ISO/IEC Directives, Part 2 - Principles and rules for the structure and drafting of ISO and IEC documents, Ninth edition, 2021

[2] Chowdhary, KR1442, and K. R. Chowdhary. "Natural language processing." Fundamentals of artificial intelligence (2020): 603-649.

[3] A. Géron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.", 2022.

[4] F. Fabbrini, M. Fusani, S. Gnesi and G. Lami, "The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool," Proceedings 26th Annual NASA Goddard Software Engineering Workshop, Greenbelt, MD, USA, 2001, pp. 97-105, doi: 10.1109/SEW.2001.992662.

[5] Z. Zhang, and L. Ma, "Using machine learning for automated detection of ambiguity in building requirements." EC3 Conference 2023. Vol. 4. European Council on Computing in Construction, 2023.

[6] A. Fantechi, S. Gnesi, and L. Semini, "Rule-based NLP vs ChatGPT in ambiguity detection, a preliminary study." CEUR Workshop Proceedings. Vol. 3378. CEUR WS, 2023.

[7] J. Bernard, Python data analysis with pandas. In Python Recipes Handbook: A Problem-Solution Approach (pp. 37-48). 2016. Berkeley, CA: Apress.

[8] W. A. Qader, M. M. Ameen and B. I. Ahmed, "An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges," 2019 International Engineering Conference (IEC), Erbil, Iraq, 2019, pp. 200-204, doi: 10.1109/IEC47844.2019.8950616.

[9] T. Zhang. "Solving large scale linear prediction problems using stochastic gradient descent algorithms." Proceedings of the twenty-first international conference on Machine learning. 2004.

[10] N. Fenton, M. Neil. “A Strategy for Improving Safety Related Software Engineering Standards”, IEEE Transactions on Software Engineering, Vol. 24, pp. 1002-1013 (1998).

[11] I. Biscoglio, M. Fusani. “Analyzing Quality Aspects in Safety-Related Standards”, NPIC 2010, 7th International Topical Meeting on Nuclear Plant Instrumentation and Control, November 7-11, Las Vegas, Nevada, USA, 2010.

[12] I. Biscoglio, A. Coco, M. Fusani, S. Gnesi, G. Trentanni. “An approach to Ambiguity Analysis in Safety-related Standards”, In Procedings of QUATIC 2010, Porto, Portugal, pp. 461-466, 2010.

[13] M. Fusani, G. Lami. “On the efficacy of safety-related software standards”, AESSCS Workshop – 10th European Dependable Computing Conference (EDCC'2014) Newcastle Upon Tyne, 13 May 2014.

[14] A. Ferrari, M. Fusani, and S. Gnesi. “Are Standards an Ambiguity-free Reference for Product Validation?”, RSSRail Conference 2017 on Reliability, Safety and Security of Railway Systems: Modelling, Analysis, Verification and Certification, Pistoia, Italy, November 14-16, 2017.

[15] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.