Detecting Constraints and their Relations from Regulatory Documents using NLP Techniques

Karolin Winter, Stefanie Rinderle-Ma

Faculty of Computer Science, University of Vienna, Vienna, Austria
{karolin.winter, stefanie.rinderle-ma}@univie.ac.at

Abstract. Extracting constraints and process models from natural language text is an ongoing challenge. While the focus of current research is merely on the extraction itself, this paper presents a three step approach to group constraints as well as to detect and display relations between constraints in order to ease their implementation. For this, the approach uses NLP techniques to extract sentences containing constraints, group them by, e.g., stakeholders or topics, and detect redundant, subsuming, and conflicting pairs of constraints. These relations are displayed using network maps. The approach is prototypically implemented and evaluated based on regulatory documents from the financial sector as well as expert interviews.

Keywords: Compliance, Regulatory documents, Requirements extraction, Text mining, NLP

1 Introduction

Extracting norms and business rules as well as process models from natural language text is an ongoing challenge since non-compliance to laws or regulatory documents can cost billions of dollars [14]. The growing amount of regulatory documents and the need to constantly update and compare already existing rules is exacerbating the situation.

Several approaches have been presented for supporting this challenge (e.g., [3, 5, 10, 18]), but they either impose restrictions on the input text (e.g., it is assumed that the text does only contain process relevant information) or produce models and rules that are incomplete or contain conflicts [22]. Moreover, the main focus of these approaches is mostly on extracting constraints or mapping them to formal rules. However, users still need to understand the rules as well as dependencies between constraints in order to be able to implement them correctly [23]. An identification of redundant, subsumed or conflicting constraints could avoid additional or unnecessary implementation effort as well as implementation errors. Another difficulty is the fact that in large companies not every constraint affects every department or stakeholder, consequently not every person has to always read every part of a document.

Grouping constraints based on *constraint related subjects*, which are for example topics, stakeholders or departments, as well as displaying relations between constraints should therefore be supported conceptually and by suitable tools.

In [26] we presented a method that is capable of identifying sentences containing constraints based on standard text mining tools as well as grouping of document fragments having similar topics or stakeholders by using term frequencies and k-means clustering. For the presented first case study on ISO security documents it was possible to identify and group fragments dealing with different topics, e.g., measurement and evaluation of ISMS or legal concerns. However, using term frequencies to group documents or sentences, even though this procedure is widely applied in the field of text mining, is rather limited and can lead to vague or incomplete results.

In this paper, we want to overcome this issue by providing means to either integrate additional information such as organizational charts or exploiting the part-of-speech tags of sentences leading to a more purposeful grouping of sentences containing constraints. Moreover, in order to tackle the lack of managing redundancies, subsumptions as well as conflicts between constraints, the method is further extended by an identification and visualization of these relation types.

For this purpose, the following research questions are stated

RQ1 How to group elicited constraints based on constraint related subjects like stakeholders or topics?
RQ2 How to identify relations between constraints?
RQ3 How to display the elicited constraints and the derived relations?

In order to answer these questions, this paper presents a method that makes use of NLP techniques and provides means for integration of additional information whenever it is available. Moreover, a definition of redundancy, subsumption and conflict of rules with respect to natural language text is stated which can be used to point out potential modelling and implementation errors.

The remainder of the paper is organized as follows. Sect. 2 describes the method, Sect. 3 the prototypical implementation which is used in Sect. 4 to evaluate the approach on a set of regulatory documents from the financial sector. A short discussion of the approach is presented in Sect. 5, followed by related work in Sect. 6. The paper concludes in Sect. 7 with a summary and outlook of future work.

2 Overall Method

In this section the method is outlined. Since it can be carried out with any NLP framework, details on our prototypical implementation are separately described in the next section. The method (cf. Fig. 1) can be divided into the three typical stages of data mining, **pre-processing**, **processing** ($\mapsto$RQ1 & RQ2) and **post-processing**($\mapsto$RQ3), each of them consisting of several steps. Pre-processing and parts of the processing steps can be viewed as a tool chain since they rely on state-of-the-art NLP techniques and data mining algorithms. The second part

of the processing and the post-processing step form the main contribution by providing a definition and application of characteristics of redundant, subsumed, and conflicting constraints with respect to natural language text.

During each stage the elicited sentences containing constraints are maintained as such and not yet mapped to a formal language. The advantage is that non expert users have a better chance of understanding the rules and their relations. After all relations have been resolved they are visualized using a graph-based representation.

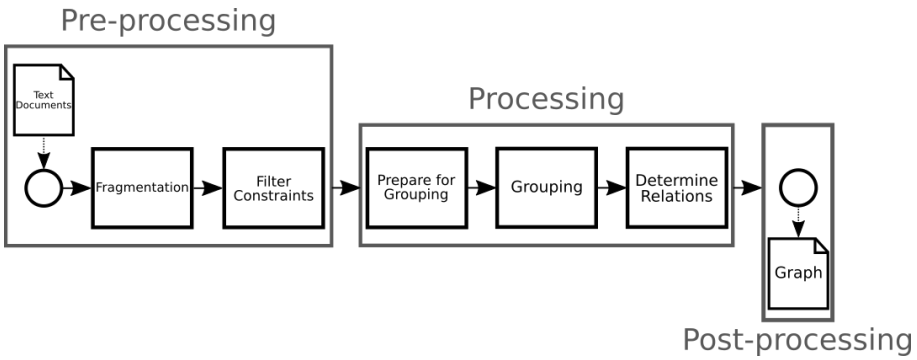


Fig. 1: Overall Method

To illustrate the method a running example is provided which is based on parts of two documents from the ISO 27000 security standard family (ISO 27001 and ISO 27011) as depicted in Fig. 2. They were used for a case study in [26]. Sentences 1-9 are taken from ISO 27000 which is an overview document and sentences 10-18 stem from ISO 27011 which outlines security topics for the telecommunications domain.

2.1 Pre-processing

First of all, each document needs to be prepared, i.e., table of contents and references are removed, since these parts do not contain valuable information on constraints. Moreover, since the documents are chunked into sentences these parts of documents would cause errors during the part-of-speech (POS) tagging process. Depending on the documents it might be necessary to remove even more parts that do not contain constraints, e.g., introductions. In addition, it should be checked whether information from tables and pictures is needed and was parsed correctly. A manual inspection of these steps might be required depending on the framework that is used. Another challenge is how to proceed with, e.g., footnotes. If a footnote, contains an explanation or a link to another document, it could be included in the final visualization. Since text passages are often related and depend on each other it might happen that one sentence refers to the subject

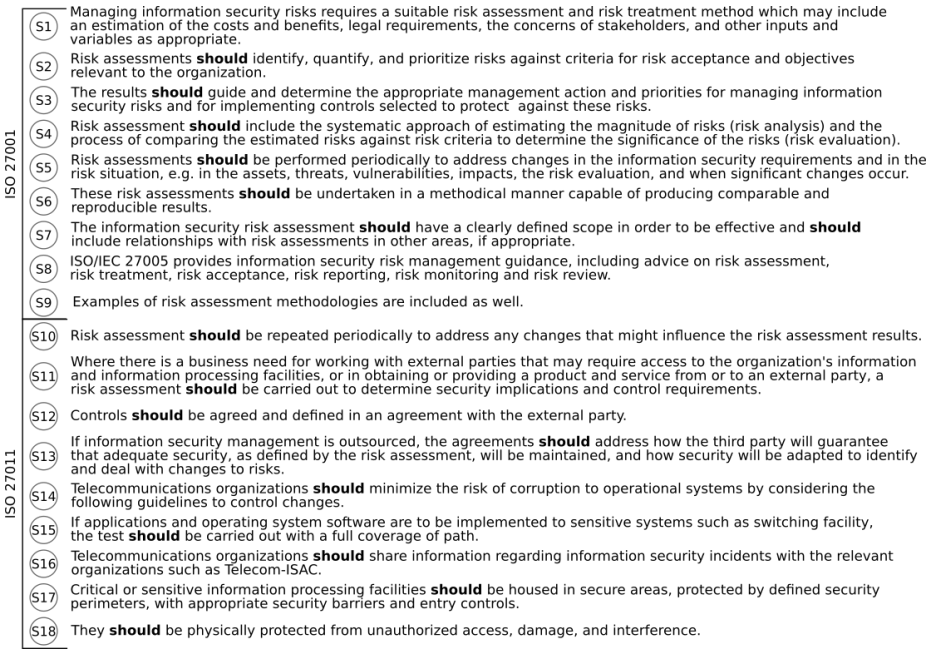


Fig. 2: Running Example–textual input

of a preceding one. In this case determiners or pronouns, e.g., `they` are used. During the preparation of the documents each of these words must be replaced by the corresponding subject of its preceding sentence. In the running example sentences $\mathcal{S}17$ and $\mathcal{S}18$ represent such a situation. Here, `they` in $\mathcal{S}18$ must be replaced by `information processing facility` from $\mathcal{S}17$. Another issue is to detect whether multiple subjects are present in one sentence. In this case, the corresponding sentence is split, resulting in two partial sentences. In the running example, $\mathcal{S}7$ is not split because the sentence does only contain one subject, i.e., `information security risk assessment`. The prepared documents form the input, so-called text corpus, for the subsequent steps.

Now, each document in the text corpus is fragmented (chunked) into sentences and POS tagged.[1] Afterwards all sentences containing constraints are filtered out. For this purpose, each sentence is scanned for markers such as `shall`, `should` or `must`. We use markers for deriving constraints, like [10] use markers for detecting BPMN elements. In addition, during the evaluation an expert interview confirmed these assumptions. If a sentence contains at least one of these markers it is tagged as constraint and the following definition can be stated.

Definition 1. *Let $\mathcal{S}$ be a set of sentences. A constraint is an element $s \in \mathcal{S}$ such that at least one marker is contained in s. The set of all constraints is called $\mathcal{C}$.*

[1] The POS tags are necessary at a later stage of the method.

In the running example, constraints are the sentences containing words (markers) written in bold font.

Sometimes, constraints are pre- or succeeded by sentences only containing explanatory information on rules but no markers. These sentences are not included in the following steps but can be included in the visualization if necessary. In the running example, these are the sentences containing no word in bold font ($\mathcal{S}1$, $\mathcal{S}8$, $\mathcal{S}9$). During the processing of sentences lemmatized words are used, in order to prevent that, e.g., plural and singular forms of nouns form different groups.[2] The final visualization still contains the original sentences.

2.2 Processing

The processing stage is divided into three steps, the preparation for the grouping, the grouping itself and the determination of relations between pairs of constraints. Consequently, the result of the processing is on the one hand a grouping of constraints and on the other hand the detection of redundant, subsumed and conflicting constraints. Three possibilities for carrying out these steps are included in order to ensure that an analyst can choose the mean that is most suitable for the given collection of documents.

Preparation and Grouping

Term Frequencies: The first option corresponds to unsupervised grouping of sentences using k-means as it was outlined in [26] and should be applied on a large collection of documents. For this, term frequencies need to be determined which can be computed by different measures. If the text corpus contains documents (or in this case sentences) that strongly vary in length, term frequency inverse document frequency (cf. [1]) is recommended resulting in a term-sentence matrix which is used for grouping the sentences.

Besides choosing a suitable term frequency measure another challenge is to determine the appropriate number of groups for k-means. Commonly applied methods for selecting the number of groups are, e.g., elbow or silhouette plots. In order to further improve the approach, we decided to use k-means++ [4]. The result of k-means clustering is a grouping of sentences based on term frequencies.

For the running example the most frequent terms are: `organizations`, `risk(s)`, `assessment`, `telecommunications` and performing k-means++ with $k = 6$ creates the groups schematically displayed in Fig. 3.

The remaining two methods correspond to a supervised grouping of the set of sentences based on a predefined list of terms. So, the labels of groups are given beforehand and each group corresponds to one of the terms that were derived by one of the following techniques.

Structure of Sentences: The second method for grouping the sentences is based on extracting constraint related subjects in an automated way without making use of additional information but by exploiting the structure of sentences and can be applied for small or mid-size document collections. A word is

[2] The quality of the outcome of this step relies on the NLP framework that is used.

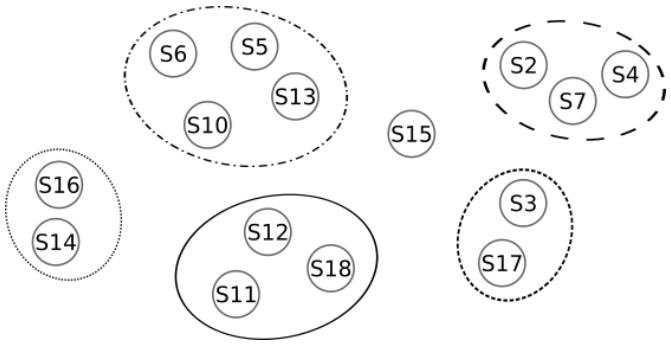


Fig. 3: Running example – grouping by term frequencies

identified as constraint related subject if it is a (compound) subject and followed by a marker, e.g., in sentence $\mathcal{S}12$ of the running example the marker is `should` and the constraint related subject is `controls`. Based on this, the list of constraint related subjects is created by examining the parse tree of each sentence and searching for the described pattern ((compound) subject + marker). In the running example constraint related subjects are, e.g., `information security risk assessment` or `information processing facility`. Since each sentence is processed in its lemmatized form, `risk assessment` and `risk assessments` are treated as one grouping subject. Terms like `information security risk assessment` and `risk assessment` are not aggregated since these might relate to different things, e.g., `risk assessment` could be another type of risk assessment than `information security risk assessment`. For the running example constraint related subjects are `risk assessment`, `information processing facility`, `telecommunication organization`, `result`, `control`, `agreement`, `information security risk assessment` and `application`.

Now, each sentence is parsed and checked whether it contains one of the terms from the constraint related subject list. If so, it is shifted to the corresponding group. Figure 4 displays this grouping for the running example.

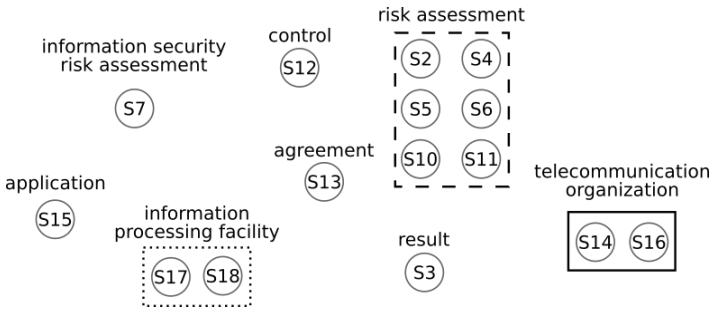


Fig. 4: Running example – grouping by sentence structure

External Information Sources: The third and last processing possibility can be used whenever external information sources, like organizational charts, glossaries, or any other knowledge provided by domain experts that contains information on how to group constraints is available. Based on this information, a list containing possible constraint related subjects is derived. Afterwards, each entry in the list is extended by synonyms. A commonly applied mean to find synonyms is to use a lexical database such as WordNet [17]. Synonyms are relevant in this case because of the diversity of language, e.g., one subject could be represented by several different words. In addition, all subjects should be lemmatized like before. The outcome is again a list containing constraint related subjects. Like in the second method each sentence is shifted into its corresponding group.

For the running example no additional information is available, therefore only the first (word frequencies) and second (sentence structure) method for grouping constraints can be applied. The third method is demonstrated in the evaluation.

Determine Relations

After grouping the set of sentences containing constraints the second part of the processing step is to retrieve dependencies between them in order to detect redundant, subsumed and conflicting constraints. To this end a classification of these types of constraints for natural language text is provided. It is based on [19] which gives a definition of these terms for constraints in a formal language. In order to transfer the characterization to sentences, we first need to define the similarity between pairs of constraint related subjects and tasks. For this Def. 2 is following the one of semantic similarity of text labels in [8]. Constraint related subjects are derived as described before ((compound) subject + marker) while for tasks the techniques of [10] can be applied, e.g., by filtering verbs. In this case we can be more precise since tasks will be preceded by markers.

Definition 2 (Semantic Similarity). *Let $\mathcal{C}$ be a set of constraints, $c_1, c_2 \in \mathcal{C}$ and let $\mathcal{W}$ be the set of all words contained in c_1, c_2. Moreover, let $\mathcal{R}$ be the set constraint related subjects of c_1, c_2, $w : \mathcal{R} \mapsto \mathcal{P}(\mathcal{W})$ be a function that separates an element in $\mathcal{R}$ into words. Let $w_1 = w(r_1), w_2 = w(r_2)$ and w_i, w_s be the weights associated with identical words and synonymous words, respectively. The semantic similarity of two constraint related subjects $r_1, r_2 \in \mathcal{R}$ is defined as*

$$sem(r_1, r_2) := \frac{2 \cdot w_i \cdot |w_1 \cap w_2| + w_s \cdot (|s(w_1, w_2)| + |s(w_2, w_1)|)}{|w_1| + |w_2|},$$

with $s(w_1, w_2)$ being the set of synonyms of w_1 that appear in w_2.

The semantic similarity of tasks t_1 of c_1, t_2 of c_2 can be defined analogously.

For the running example take $r_1 =$ `information security risk assessment` and $r_2 =$ `information processing facility`. It holds $w_1 =$[`information`, `security`, `risk`, `assessment`] and $w_2 =$[`information`, `processing`, `facility`]. Consequently, with $w_i = 1, w_s = 0.75 : sem(r_1, r_2) = \frac{2 \cdot 1 \cdot 1 + 0.75 \cdot (0+0)}{4+3} \approx 0.286$. So, these constraint related subjects do not have a high similarity.

For defining the targeted relation types of constraints, on the one hand the similarity of sentences and on the other hand a characterization of conflict between sentences is needed. While computing similarity of text is a frequently studied part of natural language processing, to the best of our knowledge, determining conflicting text parts has not been examined very well by now ([13,15]). Mostly, these approaches search for negations or antonyms. Searching for negations might not be that useful when considering constraints since these will not be stated explicitly in a regulatory document. Antonyms in this case correspond to, e.g., constraint related subjects having a low similarity score. Consequently, the following definitions can be stated.

Definition 3 (Constraint Characterization). *Let $\mathcal{C}$ be a set of constraints, $c_1, c_2 \in \mathcal{C}$ and $sim : \mathcal{C} \times \mathcal{C} \mapsto \mathcal{I}$ be a function that determines the similarity between two constraints with $\mathcal{I} \subseteq \mathbb{R}$ an interval. Let $\tau \in \mathcal{I}$ be a constant, such that $sim(c_1, c_2) > \tau$. The constraints c_1, c_2 are called*

- *redundant, iff*
  - *they belong to the same group or for constraint related subjects $r_1 \in c_1, r_2 \in c_2$ holds $sem(r_1, r_2) > \eta_1$*
  - *and for two tasks $t_1 \in c_1, t_2 \in c_2$ holds $sem(t_1, t_2) > \eta_2$*

  with $\eta_1, \eta_2 \in \mathcal{I}$.
- *subsumed, iff they are redundant and either c_1 or c_2 contains further information related to its task.*
- *conflicting, iff either*
  - *they belong to different groups or for constraint related subjects $r_1 \in c_1, r_2 \in c_2$ holds $sem(r_1, r_2) < \mu_2$*
  - *and for two tasks $t_1 \in c_1, t_2 \in c_2$ holds $sem(t_1, t_2) > \mu_1$*

  with $\mu_1, \mu_2 \in \mathcal{I}$ or they are redundant but contain different time spans.

Definition 3 of redundant, subsumed, and conflicting constraint pairs is based on a similarity function *sim* which operates on constraints that are reflected by sentences. The similarity of sentences is computed within and across each group.

For this, all words need to be represented by word vectors. For computing these vectors, several approaches have been proposed during the last years (e.g., [16,21]). In order to deliver reasonable results mostly large data collections for training the model are needed. Therefore, we suggest to use a pre-trained model or pre-trained word vectors for the language in which the documents are written. The similarity between the word vectors is then computed using a distance measure. For text mining tasks the cosine measure is a common choice [1]. The final outcome is a similarity score which is, in the case of the cosine measure, a value between -1 and 1, with 1 corresponding to absolutely similar, -1 not similar at all, i.e., $\mathcal{I} := [-1, 1]$ in this case.

After applying the characteristics set out in Def. 3 the outcome is three lists per method containing redundant, subsumed or conflicting constraints.

For the running example using **term frequencies** results in one pair of subsumed constraints, $(\mathcal{S}5, \mathcal{S}10)$ with a similarity score of ≈ 0.94031. These are obviously subsumed since the first sentence explains in more detail what

needs to be done to address changes. No redundant or conflicting constraints are found which can be easily verified by a manual inspection of the given sentences. Examples of these types are given in the evaluation.

Using the second method, i.e., **sentence structure** delivers one pair of subsumed constraints ($\mathcal{S}5, \mathcal{S}10$), no redundant and no conflicting constraints.

2.3 Post-processing

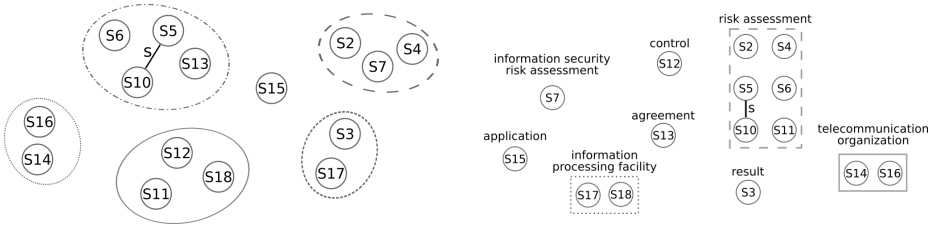
To make the derived information available for the user, a suitable visualization is needed. In this approach a graph-based structure (so-called network map, cf. Def. 4) is used but this step can be customized and any other representation could be chosen. In the network map visualization each node corresponds to a sentence and the edges indicate whether sentences are redundant, subsumed, or contradicting. Edges representing redundant and subsumed connections are labeled as `r` and `s` while contradicting ones are labeled as `c`.

Definition 4 (Constraint Network Map). *A network map is a graph* $NM = (\mathcal{C}, E)$*, with*

- $\mathcal{C}$ *being a set of nodes where each node* $c \in \mathcal{C}$ *corresponds to one constraint*
- $E \subseteq \mathcal{C} \times \mathcal{C}$ *being the edges.*

Moreover, let $w\colon E \mapsto RL := \{r, s, c\}$ *be a function assigning a label to an edge depending on the corresponding relation between the nodes that span the edge, i.e., redundant (r), subsumed (s), conflicting (c).*

Figure 5a displays the network map for the running example based on term frequencies, Fig. 5b the network map for the running example based on the sentence structure. Subsumed constraints are displayed as edges labeled `s` for subsumed. Note that no redundant or conflicting constraints were found and constraints that are not connected do not have a relation.



(a) Relations – term frequencies

(b) Relations – sentence structure

Fig. 5: Relations – running example

Another possible post-processing strategy could be to transform each sentence into a formal language in order to construct executable rules or models.

Many current approaches are capable of doing this but often conflicts, subsumptions or redundancies are not resolved correctly resulting in incomplete or contradicting rules and models. By first applying the presented approach it might be possible to resolve such clashes at all or at least in a shorter amount of time.

3 Implementation

A prototypical implementation of the method described in Sect. 2 is provided and used for the evaluation in Sect. 4. The prototype is written in Python 3 and integrates the NLP framework Spacy[3], NLTK (cf. [6]) and WordNet[4]. This decision relies on [2], which evaluated several state-of-the-art NLP frameworks.

First of all, the documents are transformed and prepared as described in 2.1. Due to the variety of document formats it is difficult to provide a generic implementation and further elaborating on this is beyond the scope of the paper.

The first generic step which is carried out using Spacy is the chunking, parsing, POS tagging and lemmatizing of sentences. As recommended, the large model for the English language is used. There are three means on which the grouping can be based on and which are applied after filtering the POS tagged sentences for markers.

The first technique uses clustering based on **term frequencies** in combination with the k-means++ algorithm and can be applied if no additional information is available and the set of documents is large. Since this correlates to parts of the method presented in [26], the corresponding parts of the implementation were migrated from R to Python 3 and integrated into the recent implementation. In the implementation TfIdf as well as k-means++ are taken from the popular scikit-learn library [20].

For the remaining two techniques the grouping relies on a list of constraint related subjects.

The second method exploits the **sentence structure** in order to derive such a list. In particular the annotation attributes of Spacy Tokens are used. Note, that during this process there are some properties that need to be taken care of, e.g., compound terms must be considered as one constraint related subject or plural and singular forms of terms should not form separate groups. For this, POS and dependency tags are considered. To enhance the performance, the list creation and shifting of sentences to their corresponding group is combined.

Extraction of constraint related subjects can also be based on **external information** (in the evaluation an organizational chart is used) and integration of a lexical database for finding synonyms which is in our implementation WordNet. Since, Spacy has no integration of WordNet, we rely on NLTK for this step. An initial list is created from the external source and every term is extended by its synonyms present in WordNet. Now that the list is given, each sentence is processed again and shifted to its corresponding group.

[3] https://spacy.io
[4] https://wordnet.princeton.edu/

The last step of the processing stage relates to finding relations between pairs of constraints. As outlined in Def. 3, the similarity between sentences needs to be computed. For this task it is either possible to train own word vectors or to use Spacy's similarity function which uses the cosine metric and word vectors that were trained with the word2vec algorithm family [16]. Since this is a pre-computed model it might happen that not every term has a vector representation, so this needs to be checked and adapted if necessary.

After computing the similarity, all sentences above a certain threshold are further examined whether they have the described characteristics for redundancy, subsumption or conflict stated in Def. 3. Automating the detection and comparison of time spans is the main challenge here. Simple functions like `isdigit()` are not sufficient since digits can also be written out.

In the last step of the method, the results are displayed as network maps. For drawing these graphs the NetworkX (cf. [12]) package is used. Each constraint is integrated and colored based on its group. The edges are drawn whenever a relation between a pair of constraints exists and labeled accordingly. For large documents the result needs to be scalable and it should be possible to display only particular groups.

4 Evaluation

For evaluating the approach a set of documents from the financial sector is used and an expert was consulted in order to estimate the results. The first document is the *BCBS 239*[5] which provides guidelines on risk management of financial institutes. The second document is the *Regulation 2016/867*[6], which specifies guidelines for credit management.

For gaining an overview of the documents and getting to know their structure an expert interview was conducted first. The expert emphasized that constraints always contain markers like `shall`, `should` and `must`. For testing the third processing option (external information sources) an organizational chart of the experts company is intergrated.

Before starting the analysis, several questions were stated, e.g., *Did the approach find sentences which do not contain constraints?* or *Were the relations between constraints correctly drawn, i.e., how precise is the approach?*.

The precision can be measured by the ratio between the number of the intersection of relevant sentence pairs and all retrieved sentence pairs divided by the number of retrieved sentence pairs,

$$Precision = \frac{|\{relevant\ pairs\} \cap \{retrieved\ pairs\}|}{|\{retrieved\ pairs\}|}.$$

Relevant in this case means, that a pair is a pair that is indicated by the domain expert to be in the correct group of relations.

[5] https://www.bis.org/publ/bcbs239.pdf
[6] https://eur-lex.europa.eu/eli/reg/2016/867/oj

Both documents are given in PDF format, and thus first of all transformed into plain text format. Afterwards, the table of contents and references, as well as the introductions are removed and each document is fragmented into sentences and POS tagged. Constraints are filtered out using markers and lemmatized for the processing stage. Each of the three methods is applied on the resulting aggregated set of constraints and the thresholds are set to $\tau = 0.97, \eta_1 = 0.8, \eta_2 = 0.5$. **Term Frequencies:** Choosing $k = 15$ results in clusters containing between 6 and 38 constraints.

The number of redundant constraints is 42, among which 10 have a similarity score of 1.0. This is due to the fact that the sentence `In the case of natural persons being affiliated with instruments reported to AnaCredit, no record for the natural persons must be reported.` appears five times in *Regulation 2016/867* in five different sections. Another example of a redundant pair of constraints is

- `If a change takes place, the records must be updated no later than the monthly transmission of credit data for the reporting reference date on which the change came into effect.`
- `If a change takes place, the records must be updated no later than the monthly transmission of credit data for the reporting reference date on or before which the change came into effect.`

with a similarity score of ≈ 0.99897. This pair could also be viewed as subsumed but the difference is so little that the approach detects a redundancy in this case, which is fine according to the consulted domain expert.

The number of subsumed constraint pairs is 10. An example is

- `Supervisors should have and use the appropriate tools and resources to require effective and timely remedial action by a bank to address deficiencies in its risk data aggregation capabilities and risk reporting practices.`
- `Supervisors should require effective and timely remedial action by a bank to address deficiencies in its risk data aggregation capabilities and risk reporting practices and internal controls.`

with a similarity score of ≈ 0.98628.

In addition, 6 conflicting pairs of constraints are retrieved, e.g.,

- `For observed agents that are resident in a reporting Member State, NCBs shall transmit monthly credit data to the ECB by close of business on the 30th working day following the end of the month to which the data relate.`
- `For observed agents that are foreign branches not resident in a reporting Member State, NCBs shall transmit monthly credit data to the ECB by close of business on the 35th working day following the end of the month to which the data relate.`

with a similarity score of ≈ 0.99669. Having a closer look at this pair of constraints and also according to the expert, revealed that this is not a conflicting constraint pair. It rather indicates a decision, i.e., whether an observed agent is resident in a reporting member state or not which must be considered by a user who wants to implement these rules. This corner case is difficult to detect by automated approaches because the conflict of time intervals is refuted by the opposite subjects indicated by a negation.

Structure of Sentences: The approach delivers 56 constraint related subjects forming also 56 groups which contain between 1 and 22 sentences.

It can be recognized that lemmatization of words did not work out entirely, since, e.g., `bank` and `banks` formed separate groups. In addition, an exceptional case can be seen. `Procedures should be in place to allow for rapid collection and analysis of risk data and timely dissemination of reports to all appropriate recipients. This should be balanced with the need to ensure confidentiality as appropriate.` In this case `this` refers to the preceding sentence as such.

Redundant pairs of constraints have similar lemmatized constraint related subjects and similar lemmatized tasks. The approach yields 42 of these, e.g.,

- `4.4 The records must be reported no later than the monthly transmission of credit data relevant for the reporting reference date on or before which the instrument was registered in AnaCredit.`
- `If a change takes place, the records must be updated no later than the date of the monthly transmission of credit data that is relevant for the reporting reference date on or before which the change came into effect.`

having a similarity score of ≈ 0.97818. The pairs differ slightly from the ones detected by the previous method. The redundant constraint pairs with similarity score 1.0 which were found using term frequencies are not present in this set. The pattern ((compound) subject + word) fails in this case and therefore the sentence is not considered anymore. One possibility to tackle this issue might be to introduce a group "undefined".

Two subsumed pairs of constraints are found, e.g.,

- `Reports should include an appropriate balance between risk data, analysis and interpretation, and qualitative explanations.`
- `Reports should reflect an appropriate balance between detailed data, qualitative discussion, explanation and recommended conclusions.`

having a similarity score of ≈ 0.97119.

The same conflicting pairs of constraints like before are obtained.

External Information Sources: To apply the third method, an organizational chart is used for manually deriving the list of constraint related subjects. It is is used for grouping the sentences and consists in this case of 15 terms (before it is extended by synonyms). Such an organizational chart contains a graphical

representation of the relation of one official and its department to another within an organization. Consequently, the grouping is structured among departments.[7] If a sentence cannot be assigned to one of the terms from the list it is shifted to a default group. Altogether, 7 groups of size 1 to 131 are received whereupon the default group is the largest one.

This method delivers 56 redundant constraints, 14 subsumed and the same 6 conflicting constraints as before. The redundant ones are the same when combining the previous two approaches. A difference can be seen regarding the set of subsumed constraints. In this case four constraints that are not present in the previous sets are given, e.g.,

- `A banks risk data aggregation capabilities should ensure that it is able to produce aggregate risk information on a timely basis to meet all risk management reporting requirements.`
- `Risk management reports should be accurate and precise to ensure a banks board and senior management can rely with confidence on the aggregated information to make critical decisions about risk.`

with a similarity score of ≈ 0.97116.

Finally, the quality of the overall method and of every processing option needs to be assessed. For the overall method, it can be stated that every sentence that was marked as constraint truly is a constraint, so the approach did not deliver false positives. What can be taken into account for comparing the three processing strategies is, e.g., the number of created groups. A large number of groups enables a differentiated view on the data but can be to fine-granular, e.g., the second method delivered groups containing only one sentence. On the other hand, the third method created few groups but these are not very distinctive. Therefore, a good balance between the number of groups and the therein contained sentences should be targeted. The first method fulfills this criterion best. For estimating the quality of the derived relations the precision scores for each method are summed up in Tab. 1.

	redundancy	subsumption	conflict
term frequencies	69% (77%)	50% (65%)	0%
sentence structure	54% (60%)	100% (100%)	0%
external information	64% (69%)	50% (61%)	0%

Table 1: Precision scores

A domain expert checked each detected pair and decided whether it is in the right category or not. Some sentences were half half, i.e., they can be partly seen as redundant or subsuming. The score in brackets indicates this by weighting these sentences in the computation with 0.5, while the other score counts them as false positives and is therefore a bit lower. The overall outcome is, compared

[7] Another possibility is to use a glossary and carry out the grouping based on the therein contained terms.

to state-of-the-art approaches fine, when considering that no restrictions were imposed on the input text (for more details cf. [24]). The precision of `conflict` is 0% because of the before mentioned corner case of two conflicting characteristics that cancel out. Moreover, the expert indicated that no conflicts are present in the documents. Conflicts might arise when updated versions of a document are considered, i.e., a new rule causes a conflict compared to an old one. Evaluating the approach on such a set of documents is planned as future work.

Post-processing:

The results for the last method (external information) are schematically visualized in Fig. 6 to demonstrate how a user could benefit from the derived results.

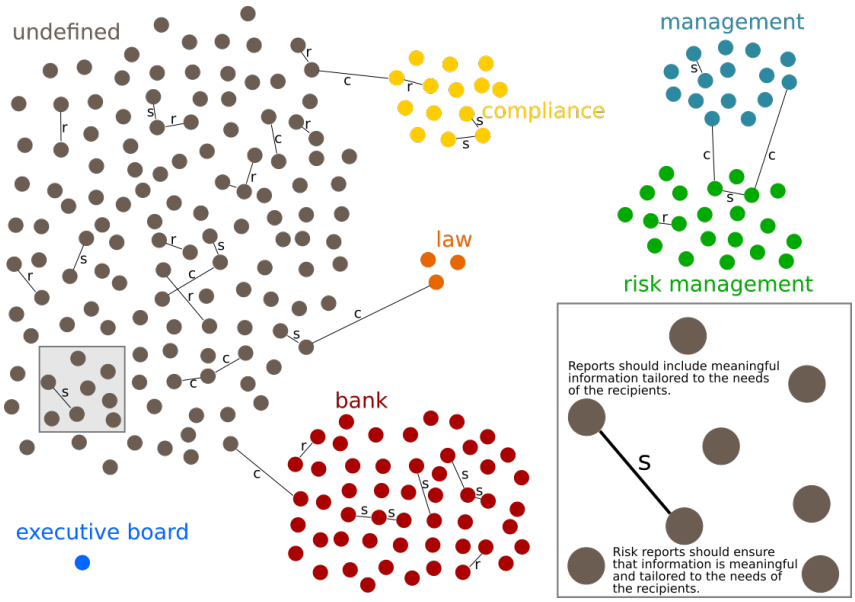


Fig. 6: Visualization – results external information

The grouping of constraints is reflected by the different colors of nodes and their regional proximity. A user can now select the subset of constraints that he wants to have a closer look at. Moreover, he can view the relations and sentences in more detail as indicated by the box in the right lower corner.

5 Discussion and Limitations

Ambiguity of language: Since natural language can be versatile the completeness of markers is hard to estimate. Also, extracting synonyms can be a

challenge since meanings of words differ when the context is changing. Using domain specific ontologies might overcome this issue. During the evaluation we also observed that, e.g., searching for time spans is not straight forward and several iterations of implementations are needed in order to get reasonable results.

Choosing a NLP framework: There are lots of NLP frameworks available and many of them provide different features. The quality of the results relies on their capabilities to parse information correctly. As it could be demonstrated in the evaluation, lemmatization was not carried out correctly for each case. (Manual) adoptions tailored to the regarded document collection might be necessary.

Integration of external information: Another task that requires manual inspection is the integration of external information for deriving the list of constraint related subjects. Again, this step relies on the input format and tools that are used and is therefore difficult to automate.

Selection of thresholds: A common challenge in data mining applications is the selection of parameters and thresholds. This approach is no exception.

6 Related Work

Most approaches in the business process compliance domain focus on creating business process models from natural language text but not on retrieving constraints as it is the target of this paper. [11] investigated BPMN model creation from text artefacts, [3] derived BPMN models based on group stories while [25] studied the creation from use cases. [10] present an approach for BPMN process model generation from natural language text which is the current state-of-the-art. The determination of UML models is targeted by [7, 18]. An approach for creating formal models for use in information systems development using the Semantics of Business Vocabulary and Business Rules (SBVR) standard is presented in [24]. Each of these approaches mostly either requires rather structured input data (sometimes combined with additional information) or produces models that lack precision.

An approach focusing on the extraction of rules is, e.g., [5] which extracts SBVR rules from natural language text but still needs a domain specific model is needed. Our approach does not require such information. [9] outline a method for extracting rules from legal documents by using logic-based as well as syntax-based patterns.

Resolving relations between sentences containing constraints is not discussed in any of the mentioned approaches but might help to improve derived business rules and process models.

7 Conclusion and Future Work

In this paper an approach for grouping sentences containing constraints and resolving relations between them was presented. Relations could be resolved based on a characterization of redundancy and conflict. A state-of-the-art NLP framework as well as common data mining algorithms were used for implementing the

method. The evaluation was carried out on a set of documents from the financial sector and the results were assessed by a domain expert.

The most crucial target of future work is to evaluate to what extent our method can resolve the lack of precision generated by state-of-the-art approaches for process rule and model extraction from natural language text. Besides that, we plan to further extend the evaluation in order to improve the implementation by covering more exceptional cases. Another interesting point is to consider sets of documents that consist of updated versions of one document and to retrieve examples of constraint pairs that changed during the versions. This might help to manage and update business rules accordingly. Creating an interactive visualization that integrates the original documents is also envisaged.

Acknowledgment

This work has been funded by the Vienna Science and Technology Fund (WWTF) through project ICT15-072.

References

1. Aggarwal, C.C., Zhai, C.: Mining text data. Springer Science & Business Media (2012)
2. Al Omran, F.N.A., Treude, C.: Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments. In: Proceedings of the 14th International Conference on Mining Software Repositories. pp. 187–197. IEEE Press (2017)
3. de AR Goncalves, J.C., Santoro, F.M., Baiao, F.A.: Business process mining from group stories. In: Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on. pp. 161–166. IEEE (2009)
4. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035. Society for Industrial and Applied Mathematics (2007)
5. Bajwa, I.S., Lee, M.G., Bordbar, B.: SBVR business rules generation from natural language specification. In: AAAI spring symposium: AI for business agility. pp. 2–8 (2011)
6. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
7. Deeptimahanti, D.K., Babar, M.A.: An automated tool for generating UML models from natural language requirements. In: Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering. pp. 680–682. IEEE Computer Society (2009)
8. Dijkman, R., Dumas, M., Van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (2011)
9. Dragoni, M., Villata, S., Rizzi, W., Governatori, G.: Combining nlp approaches for rule extraction from legal documents. In: 1st Workshop on MIning and REasoning with Legal texts (MIREL 2016) (2016)
10. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: International Conference on Advanced Information Systems Engineering. pp. 482–496. Springer (2011)

11. Ghose, A., Koliadis, G., Chueng, A.: Process discovery from model and text artefacts. In: Services, 2007 IEEE Congress on. pp. 167–174. IEEE (2007)
12. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
13. Harabagiu, S., Hickl, A., Lacatusu, F.: Negation, contrast and contradiction in text processing. In: AAAI. vol. 6, pp. 755–762 (2006)
14. Hashmi, M., Governatori, G., Lam, H.P., Wynn, M.T.: Are we done with business process compliance: state of the art and challenges ahead. Knowledge and Information Systems pp. 1–55 (2018)
15. de Marneffe, M.C., Rafferty, A.R., Manning, C.D.: Identifying conflicting information in texts. Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation (2011)
16. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 746–751 (2013)
17. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to wordnet: An on-line lexical database. International journal of lexicography 3(4), 235–244 (1990)
18. More, P., Phalnikar, R.: Generating UML diagrams from natural language specifications. International Journal of Applied Information Systems 1(8), 19–23 (2012)
19. Nguyen, T.A., Perkins, W.A., Laffey, T.J., Pecora, D.: Checking an expert systems knowledge base for consistency and completeness. In: IJCAI. vol. 85, pp. 375–378 (1985)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research 12(Oct), 2825–2830 (2011)
21. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
22. Riefer, M., Ternis, S.F., Thaler, T.: Mining process models from natural language text: A state-of-the-art analysis. Multikonferenz Wirtschaftsinformatik (MKWI-16), March pp. 9–11 (2016)
23. Rinderle-Ma, S., Ma, Z., Madlmayr, B.: Using content analysis for privacy requirement extraction and policy formalization. Enterprise modelling and information systems architectures (2015)
24. Selway, M., Grossmann, G., Mayer, W., Stumptner, M.: Formalising natural language specifications using a cognitive linguistic/configuration based approach. Information Systems 54, 191–208 (2015)
25. Sinha, A., Paradkar, A.: Use cases to process specifications in business process modeling notation. In: Web Services (ICWS), 2010 IEEE International Conference on. pp. 473–480. IEEE (2010)
26. Winter, K., Rinderle-Ma, S., Grossmann, W., Feinerer, I., Ma, Z.: Characterizing regulatory documents and guidelines based on text mining. In: Panetto, H., Debruyne, C., Gaaloul, W., Papazoglou, M., Paschke, A., Ardagna, C.A., Meersman, R. (eds.) On the Move to Meaningful Internet Systems. OTM 2017 Conferences. pp. 3–20. Springer International Publishing, Cham (2017)